

Debreceni Egyetem
Informatikai Kar

FORRASZTÁSOS KÖTÉSEK AUTOMATIUS OPTIKAI VIZSGÁLATA

Témavezető:

Dr. Cserháti Csaba

egyetemi docens

DE TTK

Szilárdtest Fizika tanszék

Készítette:

Gyarmati Gábor

Mérnök-Informatikus hallgató

Debrecen

2010

Tartalomjegyzék

Bevezetés	3
1. Forrasztások	5
2. Automatizált tesztelő berendezések	9
3. Képfeldolgozás	19
4. A programról	30
Összefoglalás	51
Köszönetnyilvánítás	54
Függelék	55
Irodalomjegyzék	59

Bevezetés

Szakdolgozatom témájául a forrasztásos kötések automatikus optikai vizsgálatát választottam. Az optikai vizsgálatok képfeldolgozási műveleteken alapulnak. Hozzám igazán közel a képalkotás áll, abban egy sajátos gyakorlattal is rendelkezek. A vizuális témakörről szerettem volna egy átfogóbb képet kapni, és azt minél több irányból körbejárni. Ismereteim bővítésére kiváló lehetőséget nyújt ez a szakdolgozat, melynek segítségével a képfeldolgozás világában is elmélyíthetem tudásomat.

A különféle forrasztási technikák során létrejött kötések kialakulásának körülményeit tanulmányozva próbálok fényt deríteni a minőségükkel kapcsolatos tudnivalókra, ideértve a leggyakrabban előforduló hibáikat. Az elektronikai eszközök gyártása során, a legjobb minőség elérése érdekében, elengedhetetlen követelmény a megfelelő tesztelő berendezések biztosítása. Mivel az alkatrészek egyre kisebbek és összetettebbek, a régi vizsgálati módszerek már nem alkalmazhatóak az elvárt hatékonysággal.

A tanszéken egy Baross pályázat keretében elkészült egy AOI berendezés, amely az alkatrészek meglétét vizsgálta. Ennek a projektnek a továbbfejlesztéseként forrasztási kötések vizsgálatát tűztük ki célul. Ehhez új megvilágító rendszert és egy új programmodult kellett el készítenem.

Szakdolgozatom során megismerkedek a tesztelő rendszerekkel, összehasonlítva a különböző típusokat használhatóságuk szerint. Külön figyelmet fordítok az optikai ellenőrzésre, és módszereire. Az optikai rendszereknek nélkülözhetetlen része a megvilágítás. Az adott vizsgálat típusához megfelelő fényforrás kiválasztása és annak elhelyezése nagy körütekintést igényel. Ezeket a technikákat sorra véve próbálom kideríteni, hogy melyik módszer a legideálisabb a forrasztások vizsgálatának elősegítésére.

A tesztelő berendezéseknek van még egy igen fontos alkotóelemük, ami nem más, mint a vizsgálat során kapott információkat kiértékelő program. Az optikai vizsgálatot végző rendszerek működésének megértése céljából készíték egy programot, ami elvégzi az ellenőrzést a kapott képeken. A program fejlesztésére a LabVIEW bizonyul a legalkalmasabbnak, mivel azt kifejezetten mérési rendszerek kialakítására készítették. Ugyanakkor több olyan VI is található beépítve, melyek a képfeldolgozást segítik.

A program készítése során megismerkedek a képfeldolgozás alapjaival, valamint azokkal a műveletekkel, amelyek a program során felhasználásra kerülnek. Használatukon túlmenően igyekszek megérteni a működésüket is. A mögöttük rejlő matematikai összefüggésekben elmélyülve vizsgálom a képre tett hatásukat.

1. Forrasztások

1.1. Forrasztás

A forrasztás egyfajta adhéziós jellegű, oldhatatlan kötés. A folyamatban diffúziós jelenségek is részt vesznek, ezek mégsem elengedhetetlen feltételei a forraszthatóságnak. Nagy különbség a hegesztéshez képest, hogy nem magát az alapanyagot olvasztjuk meg. A művelet forrasztanyaggal történik, melynek olvadáspontja jóval kisebb a forrasztandó felület olvadáspontjánál. Ez azért fontos, mert már előre legyártott alkatrészeket akarunk összeilleszteni, és így valósítható meg a felület sértetlensége. Forrasztás során az alapfém és a forrasz közt létrejön egy úgynevezett intermetallikus réteg, ami a kötést biztosítja.

1.2. Forrasztanyagok

A forrasztanyagokat olvadáspontjuk szerint két csoportba szoktuk sorolni. Az alacsony olvadásponttal rendelkező ötvözetek lágyforrasztásoknak, míg a magas olvadásponttal rendelkezőket keményforrasztásoknak nevezzük. Lágyforrasztások közé soroljuk az ónt, a bizmutot, a kadmiumot, az ólmot, a cinket, illetve ezek ötvözeit. A keményforrasztások közé az ezüst, az arany és a réz, valamint az ezekből álló ötvözetek tartoznak.^[1]

1.3. A forrasztási folyamat feltételei

Egy megfelelő minőségű kötés létrejöttének van néhány feltétele. A forrasztanyag, a forrasztandó felület, a folyasztószer, a nedvesítés, a hőforrás és az idő mind olyan tényezők, melyekre különös hangsúlyt kell fektetni. A nedvesítés az olvadt forrasztásnak a felületen történő terülését jelenti. A forrasztandó felülethez úgy kell forrasztanyagot választani, hogy az könnyen terüljön szét rajta. A folyasztószer a felület tisztítja meg, és védik az oxidációtól. Elősegítik a forrasztásnak a felületen történő terülését, viszont a kötésben már nem vesznek részt. Mivel a forrasztanyag hő hatására olvad meg, a hőforrás biztosítása elengedhetetlen a folyamat során. A hő biztosításának módja a különböző forrasztási technikáktól függően változik. Folyasztószernek fenyőgyantát, RMA-t (Rosin Mildly Activated), „no-clean flux”-ot, és vízben oldódó szerves anyagokat használnak. Az intermetallikus réteg minősége az oldódás időtartamától és a hőmérséklettől függ. Arra kell törekedni, hogy a forrasztás a lehető legrövidebb idő alatt végbemenjen.^{[2] [3]}

1.4. Forrasztási technikák

A forrasztási technikákat két alapvető csoportra szoktuk osztani. A forrasztás történhet manuálisan, vagy a folyamat elvégzését erre a célra speciálisan kialakított gépekkel végeztetjük. A gép alkalmazása esetén kétféle megvalósítás terjedt el. Egyik az újraömllesztéses technika, a másik a hullámforrasztás.

Újraömllesztéses módszert a felületszerelt alkatrészek illesztéséhez használnak. Első lépésben a forrasztást stencilnyomással előre felviszik a felületre. Második lépésben az alkatrészek elhelyezése történik meg. Ezek után a panel egy kemencébe kerül, ahol a hő hatására megolvad a forrasztóanyag, majd lehűléskor létrejön a kötés.

A **hullámforrasztásra** a furatszerelt eszközöknél van szükség. A forrasztóhullám adja a kötéshez szükséges forrasztóanyagot és hőt is. Az alkatrészek kivezetéseit a panelen található furatokba helyezik. Ezután a panelt egy megolvasztott forrasztó tartalmazó medence fölött vezetik át, úgy hogy a forrasztóhullámok a panel alját ériék.

A nyomtatott áramköri panelek tartalmazhatnak mind felületszerelt, mind pedig furatszerelt alkatrészeket. Ilyenkor először újraömllesztést végeznek el, majd utána végrehajtanak egy hullámforrasztást is.

1.5. Ólommentes forrasztás

Mivel 2006 óta törvény tiltja az ólom forrasztáshoz való használatát, így a gyártók csak ólommentes anyagokat alkalmazhatnak. Ennek egyetlen kiváltó oka a környezetvédelem. Az ólommentes forrasztóanyagok választásánál elsődleges szempont hogy tulajdonságaik hasonlítsanak a megszokott anyagokéhoz, illetve hogy alkalmazhatóak legyenek a már használatban lévő berendezésekkel. Ilyen tulajdonság többek közt az olvadáspont, az elektromos vezetés, az eddig alkalmazott felületek jó nedvesítése, és főképp hogy olcsó legyen. Tulajdonságaiban legalkalmasabbnak az ón-ezüst-réz ötvözetet találták.^[4]

1.6. Forrasztási hibák

Forrasztás során rengeteg probléma adódhat, ennek oka a folyamat bonyolultsága. A nedvesítés kapcsán két hibát szoktak említeni. A Non-Wetting azt jelenti, hogy a forrasztandó terület egy része, vagy egésze forrasztómentes marad. A De-Wetting azt jelenti, hogy mielőtt a kötés kialakulhatna, a forrasztó visszahúzódik, és önálló kis cseppek válnak le róla. Mindkét eset a felület tisztaságának hiányából ered, vagyis szennyeződés, esetleg oxidréteg marad rajta.

1.6.1. Forrasztási jellemzők

A forrasztott kötés jó és kedvezőtlen jellemzői sokféle kombinációban és változó mértékben keveredve fordulnak elő.

Jó forrasztási jellemzők (Good Characteristics):

- jó forrasztásnedvesítés
- forraszkúp képződése közbe egyenletesen keskenyedik az alkatrész kivezetésének vége felé
- a forrasztás megfelelő mennyisége
- egyenletes és fényes megjelenés. Az ólommentes kötéseknek matt a felületük, ez a magas hőmérsékletből következik.

Kedvezőtlen jellemzők (Poor Characteristics):

- hiányos nedvesítés
- a forrasztás elégtelen behatolása az átmenőfuratba
- túl sok forrasztás
- fémes és nemfémes szennyeződések
- forrasztási híd az alkatrész kivezetések között

1.6.2. A forrasztott kötések osztályozása

A vizsgálat szempontjából a forrasztott kötések az alábbi osztályozás szerint érdemes csoportosítani:

- Kielégítő:** Az elérni kívánt állapot, a gyártás során ez a mérvadó.
- Elfogadható:** Azok a kötések, amelyek egy meghatározott mértéken belül térnek el a kielégítő kategóriától nem igényelnek átdolgozást.
- Elfogadhatatlan:** Arra az állapotra vonatkozik, amely nem dolgozható újra anélkül, hogy a hiba okát meg ne állapítsanak. Utómunka lehetséges a hiba kiértékelése után.

A termelési részlegnek meg kell vizsgálnia a forrasztási folyamatot és meghatározni azokat az elvárásokat, ami alapján megállapítható, hogy a kötés minősége az elfogadható kategóriába esik.

Az átdolgozás az elfogadhatatlannak ítélt kötések esetén merül fel. Ahhoz hogy az átdolgozás sikeres legyen, meg kell állapítani a hiba bekövetkezésének körülményeit.^[2]

1.6.3. A leggyakoribb forrasztási hibák:

- forraszanyag többlet
- törés a forrasztáson
- sűrűeffektus
- légzárványok
- lyuk vagy buborék a forrasztásban
- zárlat, forraszhíd
- forraszanyag a panelen
- lábelhajlás, megemelkedett láb^[5]

2. Automatizált tesztelő berendezések

2.1. Automatizált tesztelő berendezések típusai

ATE (Automated Test Equipment)

Az automatizált tesztelő berendezések az elektronikai tesztelés fontos részét képezik. Többféle különböző megközelítés létezik az automatizált vizsgálatok megvalósítására. Mindegyik típusnak megvannak a maga előnyei és hátrányai. Ezek a vizsgálórendszerek nagy hatékonysággal alkalmazhatóak, de csak a megfelelő körülmények között. Az ATE rendszer kiválasztásánál tisztában kell lenni a különböző típusok közti különbségekkel, és azok helyes használatával.

AOI (Automated Optical Inspection)

Az AOI-t széles körben alkalmazzák a legtöbb gyártósoron. Sebessége jóval nagyobb a manuális vizsgálaténál. Időbe kerül a rendszer felállása indításkor, de utána már gyorsan működik. Használata nagy mennyiségű termelésnél ideális. Habár az emberi beavatkozásra ritkán kerül sor, a helyes beállítása időigényes.

AXI (Automated X-Ray Inspection)

Az elmúlt években jelentősen megnőtt az AXI iránti kereslet. A röntgensugaras és az optikai vizsgálat közt sok hasonlóság fedezhető fel. Az AOI csak azoknál a paneleknel alkalmazható, melyeknél a kötések jól láthatóak. A BGA (Ball Grid Array) technika során merült fel az igény egy olyan vizsgálóberendezésre, amely az optikailag nem látható elemek ellenőrzését is lehetővé teszi. AXI-val nem csupán az alkatrészek alatti kötések vizsgálhatóak, de feltárja a forrasztásoknak azokat a hibáit is, melyek hagyományos AOI berendezéssel nem látható (pl. szennyeződés az anyagban). A röntgensugár használatának külön előnye, hogy ha több irányból is készül felvétel, akkor azokból 3D-s kép alkotható. Ezáltal kapva pontosabb és bővebb információkat a forrasztások állapotáról.

ICT (In-Circuit Test)

Az ICT-t már régóta használják nyomtatott áramköri panelek tesztelésére. A vizsgálandó panelt egy tűágyra helyezik, az alkatrészek kivezetései ezekhez a tűkhöz fogak csatlakozni. Segítségével nem csak a rövidzár, a szakadás, és a hibás alkatrészek deríthetőek fel, de az IC-k működéséről is képet kaphatunk. Hátránya hogy minden paneltípushoz külön tűágyat kell

tervezni. A kontaktustüskéket nagyon pontosan kell elhelyezni. Az elektronikai alkatrészek kivezetéseinek számának növekedése miatt mára már kevesebb helyen használják, mint régen, de még így is az egyik legjelentősebb tesztelési módszer.

MDA (Manufacturing Defect Analyzer)

Az MDA-t tulajdonképpen tekinthetjük az ICT leegyszerűsítésének. Csak a rövidzár, a szakadás és az alkatrésztérték gyártási hibáit keresi. Az ICT-nél olcsóbb, de kisebb a hibafelderítése is.

FT (Functional Testing)

A funkcionális teszt az elektromos áramkör helyes működését vizsgálja. A rendszer egy jelmintát küld a panel bemenetére. Ez gyakran egy, a valós használatban kapható jelet szimulál. A kimenőjelet egy helyes kimeneti mintával veti össze. A funkcionális vizsgálat előnye, hogy a tesztelést gyorsan képes végezni.

JTAG Boundary Scan (Joint Test Action Group)

A peremfigyelés viszonylag új módszer a vizsgálatok terén, de egyre szélesebb körben alkalmazzák. A modern áramkörökbe (pl. DSP, FPGA) már a gyártás során beépítik a tesztelést segítő eszközöket. A Boundary Scan kompatibilis IC a tesztelés során küld egy adatsort, és figyeli a kimenőjelet, majd a begyűjtött információk alapján értékeli ki az áramkör állapotát.^[6]

2.1.1. Alkalmazásuk

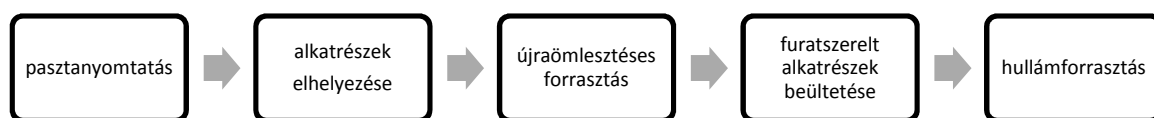
Forraszhibák esetén az AOI, az AXI és az ICT egyaránt képesek detektálni a szakadást, és a rövidzárat, forraszhidat. A hiányos forraszt és a többletet az ICT már nem veszi észre. A forraszt minőségét pedig csak az AXI tudja kiértékelni. Ide értjük a belső lyukakat és a szennyeződések.

Alkatrészhibáknál mindhárom észreveszi a megemelkedett lábat, a hiányzó-, valamint a rosszul elhelyezett alkatrészt. A hibásan működő alkatrészt viszont csak az ICT képes kiszűrni.

BGA vizsgálatához az AOI nem használható, de az AXI és az ICT kiválóan megfelel.^[7]

2.1.2. Vizsgálóberendezések elhelyezése

Egy áramköri panel gyártása több fázisból áll. Ezt most elsősorban a forrasztási folyamatok szerint tanulmányozzuk. Hiba a gyártás bármely szakaszában történhet, ami a rendszer összetettségéből adódik. Egy áramköri panel gyártási folyamata leegyszerűsítve a következőképpen néz ki:



1. ábra: áramköri panel gyártási folyamata

A tesztelő berendezéseket az egyes lépések után szokták elhelyezni. Ellenőrzést csak a teljes folyamat legvégén végezni nem érdemes. Ekkor már nagyon nehéz megállapítani a hiba bekövetkezésének az okait, valamint kivizsgálni a körülményeit. A gyártás elején keletkezett hiba továbbgyűrűzik, és akár még több hibát eredményezhet. Az szintén nem jelentene megoldást, hogy minden művelet után ellenőrzőkészülék álljon. Ennek egyik oka, hogy a berendezések rendkívül költségesek. A másik ok az, hogy egy-egy vizsgálat meglehetősen időigényes. A termelési folyamatot pedig nem szívesen lassítják a gyártók. Ugyanakkor a lehető legjobb minőségre is törekednek. Mindezek figyelembevételével szoktak egyfajta kompromisszumos megoldást keresni.

Pasztanyomtatás után a pasztavastagságot lézerrel lehet mérni.

Az alkatrészek elhelyezését AOI vizsgálhatja, de itt még nincs nagy szerepe, általában nem is igen alkalmazzák.

Annál fontosabb viszont az újraömllesztéses forrasztást követő hibakeresés. Az optikai és röntgenes berendezések egyaránt kiválóan alkalmazhatóak. Itt már mind az alkatrészeket, mind a kötéseket figyelni kell. Az időben detektált hibák most még viszonylag könnyen javíthatóak.

Az alkatrész beültetés vizsgálatára szintén AOI-t lehet használni.

A hullámforrasztás után megint csak szükséges az AOI illetve AXI vizsgálatok elvégzése.

A gyártási folyamat végén pedig az In-Circuit valamint a Funkcionális Teszt ad információt az áramkörök helyes működéséről. Ha netán egy hibát eddig nem sikerült volna detektálni, az ICT és FT még kiszűri.^[8]

2.2. Automatikus Optikai Vizsgálat

2.2.1. AOI

AOI - Automatic / Automated Optical Inspection. Ellenőrzési módszer, mely a képfeldolgozáson és a gépi látáson alapul. A kép kiértékelése közben vonja le a következtetéseket a vizsgálandó objektum állapotáról. A használt eljárások beállíthatók a feladatnak megfelelően. Elsősorban a nyomtatott áramköri paneleket, azokon belüli is az alkatrészeket és a forrasztott kötések ellenőrzi, de a gyártási folyamat során bárhol elhelyezhető. Vizsgálható vele a stencilnyomtatás és az alkatrész beültetés is.^{[9] [10]}

2.2.2. Az emberi és az automatikus vizsgálat összehasonlítása

Érvek az emberi vizsgálat mellett:

- Nem igényel különösebb befektetést (a betanítástól eltekintve).
- Az emberi tapasztalatot még egy jól programozott rendszer sem helyettesítheti.
- A vizsgálórendszer előállítása bonyolult és időigényes feladat.

Érvek az automatikus vizsgálat mellett:

- Az ember szeme hamar elfárad.
- Az ember koncentrációkéessége szintén korlátos.
- A gép gyorsabban végzi el a feladatot.

A technika fejlődésével az elektronikai alkatrészek mérete csökkent, sűrűségük nőtt. Mára ez olyan fokúvá vált, hogy nehéz hozzáférni a vizsgálatához szükséges kivezetésekhez, az emberi ellenőrzés pedig már nagyítóval is lassú és körülményes. Ugyanakkor a számítástechnikával együtt rohamosan fejlődik a gépi látás és a képfeldolgozás. Növekszik a kamerák felbontása, újfajta megvilágítási technikák jelennek meg, és egyre hatékonyabbak a képfeldolgozási algoritmusok.^[11]

2.2.3. Vizuálisan detektálható hibák:

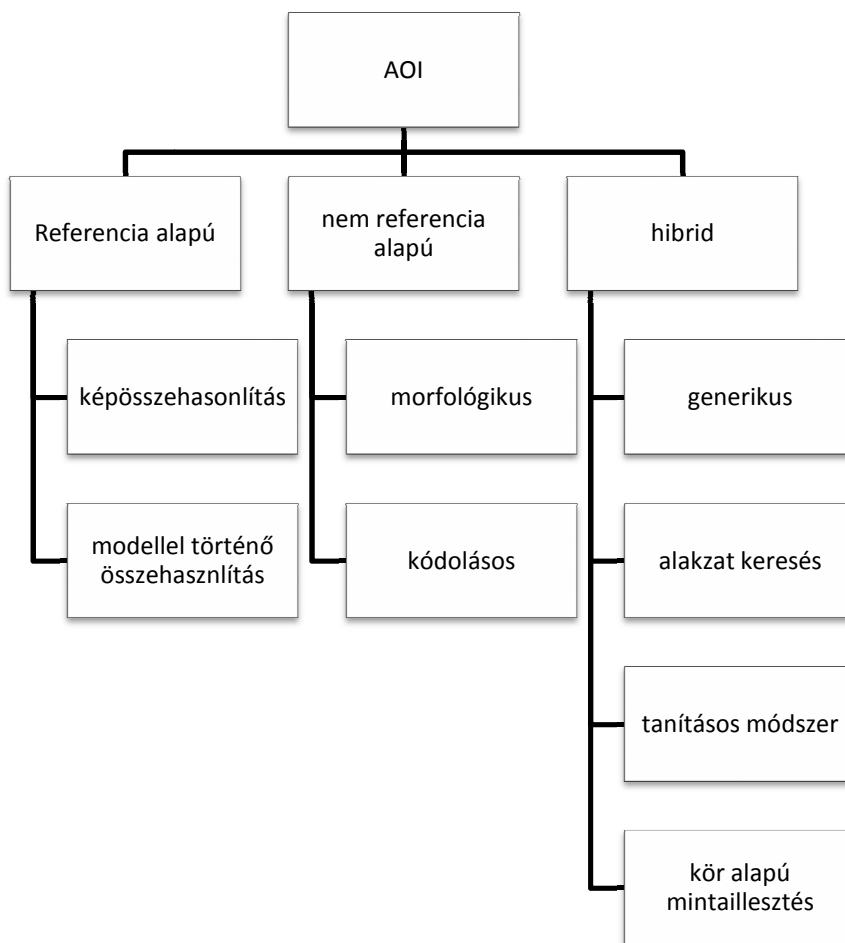
Pasztavizsgálat esetén nézzük a forraszpaszta hiányát, mennyiségét (kevés vagy túl sok) és elhelyezkedését. Észrevehető továbbá a pasztahíd, az elkenődés és a szennyeződés.

Alkatrészeknek a hiányát, pozícióját (eltolódás, elfordulás, élén-, hátán fekvés), polaritását, sérülését vizsgálhatjuk. Továbbá megfigyelhető a rosszul vagy egyáltalán nem forrasztott elem, a forraszhíd, valamint a sírkő effektus.

A ragasztónak a hiányát és a felületen történő elkenődését vizsgálhatjuk.

Ezekon kívül a hiányzó huzalvég, a rossz huzalforrasztás és a kontaktus felület nedvesítés hiányát detektálhatjuk.^[5]

2.2.4. AOI rendszerek osztályozása



2. ábra: AOI rendszerek osztályozása

A **referencia alapú eljárásnak** a lényege hogy egy referenciaképet hasonlítunk a vizsgálandó darabról készített felvétellel. Referenciának használhatunk egy tökéletes darabról (Golden Board) készült felvételt, vagy egy CAD-állományt. A módszer használható alkatrészek és forrasztások hiányának detektálására, valamint zárlatok keresésére. Hátránya hogy túl érzékeny.

A **nem referencia alapú eljárások** nem használnak sablont. Helyettük tervezési szabályok alapján történik az ellenőrzés. A módszer kiküszöböli a túlérzékenységből adódó problémákat, de nem vesznek észre olyan hibákat, melyek nem ütköznek a tervezési szabályokba.

A **hibrid eljárások** az előző két módszer előnyeit ötvözik, de ennek következtében jóval bonyolultabbak is.^{[5] [10]}

2.2.5. Vizsgálat hibái

A vizsgálat során előfordulhat olyan helyzet, hogy a rendszer nem vesz észre egy hibát. Ezt a jelenséget nevezzük átcúsúzott hibának. Ennek oka lehet az, hogy a hibás területet egy kiemelkedő alkatrész eltakarja a kamera elől. A későbbi vizsgálatok (AXI, In-Circuit, vagy funkcionális teszt) során a probléma még detektálható.

Gyakoribb eset, hogy a vizsgálat során egyes részeket hibásnak vél a rendszer, holott az teljesen megfelel a minőségi előírásoknak. Ezt hívjuk pszeudo-hibának.

Mikor a minőségi előírás által hibának tartott esetet detektál az AOI, akkor beszélünk valódi hibáról.^[9]

2.2.6. AOI rendszer elemei

A vizsgálatot számítógép vezérli. Ez irányítja a rendszert, itt találhatóak a vizsgálatához szükséges információk (pl. a CAD ábrák), valamint ide lesz mentve a vizsgálat eredménye. A tárgyasztalon helyezkedik el a vizsgálandó áramköri panel. Ennek pozícionálását a mozgatógység végzi. Az optikai fej felül helyezkedik el. Ez tartalmazza a kamerát és a fényforrást, ami a rendszer egyik meghatározó eleme.^[10]

2.3. Megvilágítás

Az optikai rendszereknél elengedhetetlen feladat a világítás biztosítása. A helyes fényforrás kiválasztásával és megfelelő elhelyezésével rengeteg felesleges munkától kímélhetjük meg magunkat. Kiemelhetünk apró, de fontos részleteket, míg a számunkra jelentéssel nem bíró tartalom eltüntethető.

2.3.1. Fényforrások

LED (Light Emitting Diode)

A fényemittáló diódák anyaguktól függően változatos spektrumon világítanak, erős intenzitással. Élettartamuk elérheti akár a 100 000 órát is.

Fluoreszcencia

Színe fehér, világossárga. Intenzitása erős. 5000-7000 órán át használható. Előnye hogy olcsó és nem melegszik.

Halogén

Színe fehér, világossárga. Nagyon erős fénnel világít. Élettartama 200 és 3000 óra közé tehető. Nem drága, viszont erősen melegszik.

Xenon

Színe fehér, világoskék. Nagyon erős a fénye. Élettartama 3000 - 7000 órán keresztül használható.

Elektrolumineszcencia

Zöld fényű, gyenge intenzitással világít 2000 és 5000 óra között. Nem melegszik.

Neon

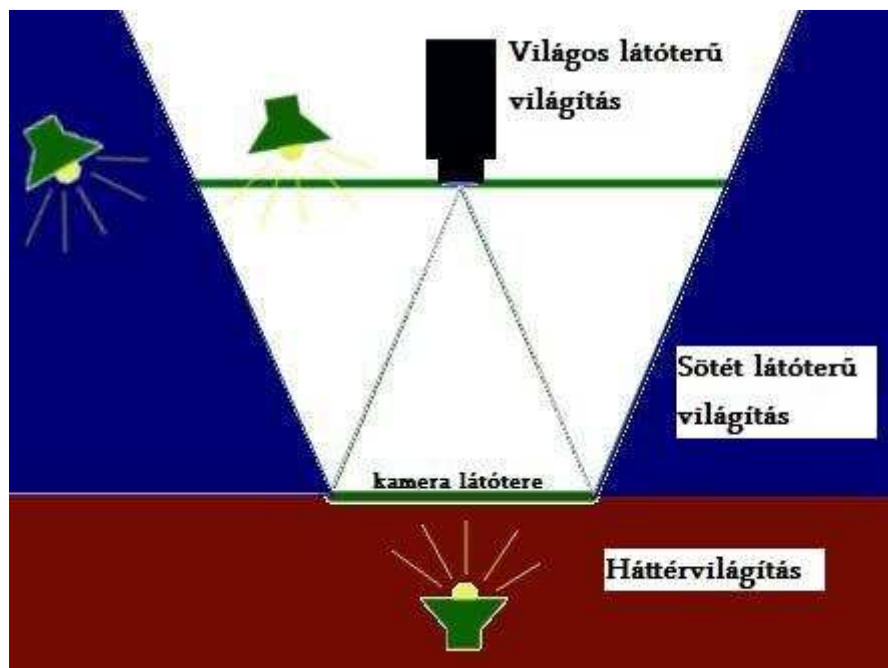
Változatos színskálán, szórt fényt biztosít. Fényerejéhez képest nem nagyon melegszik.

Ezekon kívül használnak még nagynyomású nátriumlámpát, UV-t (fekete fény), és infrát. Leggyakrabban a LED-et alkalmazzák annak kedvező tulajdonságai miatt. A fényerő és a bevilágított terület növelése érdekében többet is elhelyeznek egy csoportban.^[12]

2.3.2. A fényforrás elhelyezése

A fényforrás elhelyezésének módját a kamera látóterének szempontjából szoktuk meghatározni. Eszerint háromféle módszert különböztetünk meg egymástól:

- Bright Field (Partial vagy Full) (világos látótér)
- Dark Field (sötét látótér)
- Back Light (háttérfény)



3. ábra: megvilágítási típusok a látótér szempontjából

A kamera látóterének a kétszeresét vesszük a kamera síkjában. Ha a fényforrást azon belül helyezzük el, akkor világos látóterű megvilágításról beszélünk. Sötét látóterű megvilágításnak nevezzük azt, amikor a fényforrás ezen a tartományon kívül helyezkedik el. Háttérvilágítás esetén a fényforrás a vizsgálni kívánt tárgy mögött található.

Sötét látóterű megvilágítással a formát és a kontúrt hangsúlyozhatjuk ki. A felületen lévő kiemelkedések és mélyedések lesznek világosak, a sima részek pedig sötétek. A világos látóterű megvilágítás ezzel szemben a felület lapos részeit mutatja. A kiemelkedések és mélyedések sötétek lesznek.^[13]

2.3.3. A T-E-S-T

A számunkra optimális világítás kiválasztásában az úgynevezett „A T-E-S-T” nyújt segítséget. Az elnevezés az „Absorption”, „Texture”, „Elevation”, „Shape” és „Translucency” szavakból tevődik össze. A módszer a különböző technikákat veti össze ezekkel a jellegzetességekkel.

„Absorption” (abszorpció)

Az abszorpció az anyag fényelnyelő, -áteresztő, és -visszaverő tulajdonságát jelenti.

Drótszín, tintával nyomtatott iratok, műanyag kupakok, és UV-kibocsátás vizsgálata tartozik ide.

DOAL esetén az egységesség technikája biztosítja a megfigyelhetőséget a sima felületen történő abszorpcióváltozásnak.

Rings&Arrays használata alkalmazásfüggő.

DOME, SCDI, CDI: Egyenetlen felületen történő abszorpcióváltozás megfigyelhetőségét az egységesség technikája biztosítja.

„Texture” (textúra)

Textúra alatt a felület szerkezetét, simaságát értjük.

Lézerrel munkált felületek, dörzspapír, karcolt felszín, anyagi változások elemzésénél jellemző szempont.

DOAL: A felület világosabb a sima részekenél.

Rings&Arrays használata alkalmazásfüggő.

DOME, SCDI, CDI: a textúrát minimalizálja

Dark-Field: a textúrált felület világosabb lesz, mint a sima.

„Elevation” (kiemelkedés)

A magasságot jelenti, a felülettől a kamera felé (Z-tengely).

Bemetszett részek, domborulat, vésés, külső törmelékek vizsgálatát soroljuk ide.

DOAL: szögletes felületek sötétebbek.

Rings&Arrays használata alkalmazásfüggő.

DOME, SCDI, CDI árnyékok minimalizálására használható.

Dark-Field: világossá teszi a külső éleket.

Strukturált: a kiemelkedés változásokat mutatja.

„Shape” (alakzat)

A „Shape” az alakzatban történő változásokat jelenti az X-Y tengely mentén. Ezt hívjuk kontúrnak. Szállítószalagon lévő tárgyak, érmék, szegélyméret figyelésénél játszik szerepet.

BackLight: külső kontúrt mutat.

DOAL: A változások nyilvánvalóak, ha a háttér különböző.

Dark-Field: A kontúr hangsúlyozott, a lapos felületek sötétebbek a kiemelkedőknél.

„Translucency” (átlátszóság)

Átlátszóság alatt az anyag sűrűségfüggő fényáteresztését értjük. Fúrt lyuk, vékony terület műanyagon, műanyag lencse azonosítószáma, többrétegű anyag, hulladék folyadékban, és LCD vizsgálat a jellemző.

BackLight az átlátszó és átlátszatlan területek közti különbséget mutatja.

DOAL: Az átlátszó, lapos burkolatot minimalizálja, valamint az átlátszó és átlátszatlan részek közti különbséget mutatja, ha a háttér eltérő.

Rings&Arrays használata alkalmazásfüggő.

DOME, SCDI, CDI: Az átlátszó, egyenetlen burkolatot minimalizálja, valamint az átlátszó és átlátszatlan részek közti különbséget mutatja, ha a háttér eltérő.^[14]

2.3.4. Következtetés

Mindezeket összegezve azt a következtetést lehet levonni, hogy a forrasztások vizsgálatához a sötét látóterű megvilágítás a legmegfelelőbb. Fényforrásnak ide is a LED illik a legjobban. Egy csoportban többet is összegyűjtve úgy kell elhelyezni őket, hogy a fény egyenletes legyen.

3. Képfeldolgozás

3.1. Digitális képek

A kép az információ vizuális megjelenítésére szolgál. A digitális képek képpontokból állnak. Így tehát képnek tekinthető minden olyan kétdimenziós tömb, melyek elemei megfeleltethetők pixelértékeknek. Szürkeárnyaltos kép esetén ezek 0-tól 255-ig terjednek és a képpont intenzitását jelentik. A bináris képeknél a pixel csak 0 és 1 értékeket vehet fel. Az RGB képek egy pontjához külön meg kell adni a piros, a zöld és a kék összetevők intenzitását.

3.2. ROI

A ROI (Region of Interest) érdeklődési területet jelent. Ez tulajdonképpen a képnek az a része, amellyel foglalkozni szeretnénk. Ha egy nagy képnek csak egy kis tartománya tartalmaz számunkra értékes információt, akkor érdemes hozzá egy ROI-t definiálni. A ROI-val időt és erőforrást takaríthatunk meg. Mivel nem a teljes területen hajtjuk végre a képfeldolgozási műveleteket, így a program futási ideje töredékére csökkenhet. Másrészt pedig kevesebb memória is szükséges.

LabVIEW-ban a legegyszerűbb módja egy ROI létrehozásának, ha az ImageDisplay-hez tartozó ROI-Tools-t használjuk. Miután kiválasztottuk az alkalmazandó formát, a Display-en tudjuk kijelölni a vizsgálni kívánt területet.

ROI-t egy konstans segítségével is megadhatjuk a programban. A ROI-nak egy különösen hasznos tulajdonsága hogy könnyen konvertálható maszkká. Maszkból is egyszerűen létrehozhatunk ROI-t.

3.3. Morfológia

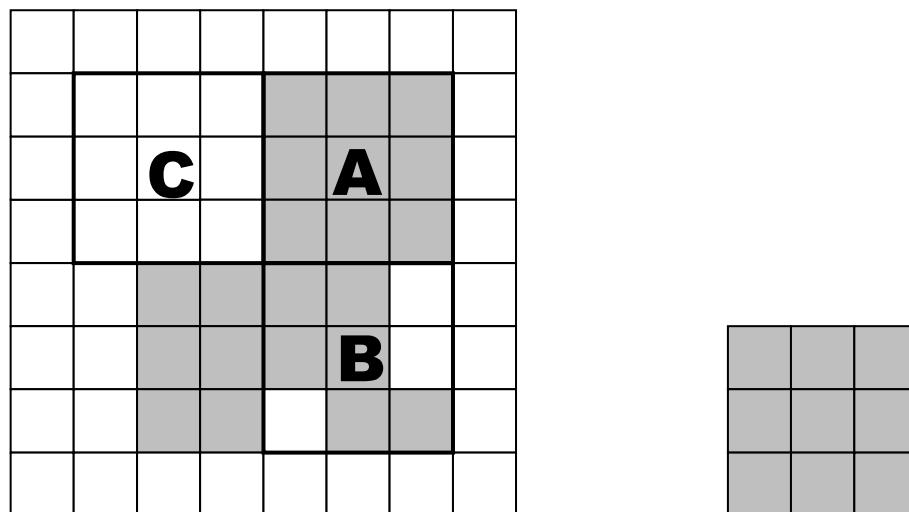
A morfológiai eljárások egy erőteljes részét képezik a nem-lineáris képfeldolgozási technikáknak. A kép tartalmazhat nem kívánt információkat, például zajokat, objektumokat, melyek a kép szélét érintik, egymást érintő részecskéket, vagy az alakzatok egyenetlen határát. A morfológiai operátorok segítségével az objektumok alakját és méretét tudjuk megváltoztatni. A módszer azon alapszik, hogy az új pixelérték függ a környező pixelektől is. Egy úgynevezett szerkezeti elemmel letapogatjuk a képet. Letapogatás során a kép és a

szerkezeti elem közti kapcsolatot értékeljük ki. A szerkezeti elem egy mátrix, mely pixelértékeket tartalmaz.

$$\begin{bmatrix} 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \end{bmatrix}$$

4. ábra: példák 5x5-ös bináris szerkezeti elemre

Miután a szerkezeti elemet ráhelyeztük a képre, az azonos pozíciókban lévő pixeleket viszonyítjuk egymáshoz. Amennyiben a szerkezeti elem 1 értékeihez a képen is 1-esek tartoznak, akkor beszélünk illeszkedésről. Ha legalább egy helyen van egyezés, akkor azt találatnak (vagy érintésnek) nevezzük. Mikor egyetlen 1-es értékű elem sem egyezik meg, akkor azt mondjuk, hogy mellé. Az alábbi ábrán szürkével vannak jelölve az 1 értékű pontok, fehérrel a 0-k.



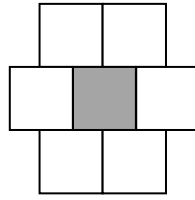
5. ábra: a jobboldalt látható szerkezeti elem a képen illeszkedik (A), talált (B), mellé (C)

A pixelek kapcsolódását tekintve megkülönböztetünk négyes- és nyolcas kapcsolódást. Négyes kapcsolódás esetén a csak az oldalukkal érintkező pixeleket tartjuk szomszédosnak. Nyolcas kapcsolódásnál elfogadjuk a sarkok érintkezését is.



6. ábra: példa négyes és nyolcas kapcsolódásra

A pixeleket tekinthetjük négyzetesnek, vagy hatszögletűnek. Hatszögletű esetén minden második sort képzeletben eltolunk egy félpixelnyivel.



7. ábra: „hatszögletű” pixel kapcsolatai

3.3.1. Morfológiai műveletek

A morfológiai eljárások két alapvető művelete az erózió és a nyújtás. A összes többi művelet ezekből eredeztethető.

Erózió (erosion)

jelölése: $g = f \otimes s$

$$g(x, y) = \begin{cases} 1 & \text{ha } s \text{ illeszkedik } f \text{ - re} \\ 0 & \text{egyébként} \end{cases}$$

$g(x, y)$ az új pixelérték, f a kép, s pedig a szerkezeti elem.

Nyújtás (dilation)

jelölése: $g = f \oplus s$

$$g(x, y) = \begin{cases} 1 & \text{ha } s \text{ érinti } f \text{ - et} \\ 0 & \text{egyébként} \end{cases}$$

Nyitás (opening)

$$g_{nyit} = (f \otimes s) \oplus s$$

A kisméretű objektumokat eltüntet, a nagyoknak pedig a szélét simítja.

Bezárás (closing)

$$g_{nyit} = (f \oplus s) \otimes s$$

Akisebb lyukakat kitölti, az egymáshoz közel elhelyezkedő objektumokat összekapcsolja, a nagyoknak a határát simítja.

Hit-Miss

$$g_{hit-miss} = (f \otimes s_{hit}) / (f \oplus s_{miss})$$

Slankítás (thinning)

$$g_{sovány} = f / ((f \otimes s_{hit}) / (f \oplus s_{miss}))$$

Erodálás történik, de aprózódás nélkül.

Hízalás (thickening)

$$g_{kövér} = f \cup ((f \otimes s_{hit}) / (f \oplus s_{miss}))$$

Az objektumokat nyújtja, de anélkül hogy a közeliak összekapcsolódnának.

Gradiens kifelé (Outer gradient)

$$g_{g-out} = f - (f \otimes s)$$

A külső kontúrt adja.

Gradiens befelé (Inner gradient)

$$g_{g-in} = (f \oplus s) / f$$

A belső kontúrt adja. ^{[15] [16]}

3.4. Korreláció

Két jel korrelációja azt határozza meg, hogy azok hol és mennyire hasonlítanak egymásra. Az egyik jel legyen egy minta, aminek a másik jellel való egyezésére vagyunk kíváncsiak. A mintát egy pontban ráillesztjük a jelre, majd elemenként összeszorozzuk a kettőt. A szorzatokat összegezzük, és ez az összeg adja meg, hogy az adott helyen mennyire pontos az egyezés. A mintát a jel minden pontjában így végignézzük, és ahol a legnagyobb értéket kapjuk, ott a legnagyobb a két jel hasonlósága. Képfeldolgozásban mintaillesztésre használják. Korrelációt konvolúció segítségével is szoktak számolni, mivel képleteik csupán egyetlen előjelben térnek el egymástól. Képlete 1D-s diszkrét esetben

$$C(x) = \sum_{m=0}^{M-1} g(m)f(x+m)$$

ahol:

- $g(m)$: a minta
- $f(x)$: az alapjel
- $c(x)$: a minta és az alapjel korrelációjának az eredménye
- M : g elemeinek a száma

Képlete 2D-s diszkrét esetben:

$$C(x, y) = \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} g(m, n) f(x + m, y + n)$$

ahol:

- $g(m, n)$ a minta
- $f(x, y)$ az alapjel
- $c(x, y)$ f és g korrelációja
- M és N sorból és oszlopból áll g

Ahhoz, hogy eredményként -1 és 1 közé eső értéket kapjunk, normalizálást kell végezni.

Normalizált kereszt-korreláció:

$$R(x, y) = \frac{\sum_{m=0}^{M-1} \sum_{n=0}^{N-1} (g(m, n) - \bar{g})(f(m + x, n + y) - \bar{f}(x, y))}{\sqrt{\sum_{m=0}^{M-1} \sum_{n=0}^{N-1} (g(m, n) - \bar{g})^2} \sqrt{\sum_{m=0}^{M-1} \sum_{n=0}^{N-1} (f(m + x, n + y) - \bar{f}(x, y))^2}}$$

ahol:

- g a minta
- f az alapjel
- \bar{g} az átlagos intenzitásértéke a mintának
- \bar{f} az átlagos értéke az aktuális tartománynak
- $R(x, y)$ g és f normalizált keresztkorrelációs értéke az (x, y) helyen^{[16][17]}

3.5. Pattern Matching

Mintaillesztéssel meghatározhatjuk egy kép azon területét, amely megegyezik egy ismert referencia mintával. Ezt a mintát szokás még modellnek vagy template-nek hívni. A „template” egy idealizált reprezentációja a képen található keresendő objektumnak. A mintaillesztéssel megtalálhatjuk a sablont a fényerőtől, az enyhe elmosódástól, zajtól és a

kiseb geometriai transzformációtól függetlenül. A mintaillesztő algoritmusoknak nagy szerepük van a gépi látásban. A leggyakoribb eljárások, amik ezen alapszanak:

- **Alignment** (középpont meghatározás)

Meghatározza a pozícióját és irányát egy ismert objektumnak. A kezdőpontot referenciapontként használja az objektumon.

- **Gauging** (mérés)

Hosszúságot, átmérőt, szöget mér. Ha a mérés nem esik a beállított toleranciatartományba, a komponens elutasított. Érdekes mintaillesztést használni a mérendő objektum helyének meghatározásához.

- **Inspection** (Vizsgálat)

Egyszerű hibák, mint például hiányzó részek vagy olvashatatlan nyomtatás detektálása.

A mintaillesztés az alkalmazásnak megadja, hogy hányszor találta meg a „template”-et a vizsgált képen, és hogy azok hol helyezkednek el rajta. Például egy nyomtatott áramköri panelen több egyforma alkatrész van jelen, és a feladatunk ezek vizsgálata.

Mivel a mintaillesztés a legtöbb gépi látáson alapuló alkalmazás első lépése, megbízhatóan kell működnie a különböző körülmények között. Egy alkatrész vizuális megjelenése esetenként teljesen eltérő lehet az elhelyezkedéséből vagy a fényviszonyokból kifolyóan.

1.) Pattern Orientation and Multiple Instances (Minta orientáció és többszörös előfordulás)

A mintaillesztő algoritmusnak még abban az esetben is meg kell találnia a „template”-et, ha az a képen elforgatva vagy skálázva van. Az eljárás detektálja a minta jelenlétét a képen, a helyét és az orientációját. Mindezt teszi a minta összes előfordulásánál.

2.) Ambient Lighting Conditions (Környező megvilágítási feltételek)

A mintaillesztő algoritmusnak képesnek kell lennie a minta megtalálására a fényerőváltozás ellenére is.

3.) Blur and Noise (Homályos és zajos körülmények)

Homályból vagy zajból kifolyólag valamilyen transzformáción átesett képen is meg kell találni a mintát. Elmosódottságot okozhat a rosszul beállított fókusz, vagy a képmező mélységének megváltozása.

3.5.1. Pattern Matching technikák

Normalized Cross-Correlation (Normalizált kereszt-korreláció)

A Normalizált Kereszt-korreláció a legáltalánosabb módja egy minta megtalálásának. Mivel az alapul szolgáló eljárás egy sor szorzás műveleten alapszik, a korrelációs eljárás rendkívül időigényes. Az eljárás sebességét növelhetjük, ha csökkentjük a kép méretét, vagy leszűkítjük a képen a keresési tartományt, amelyben a minta előfordulhat. A Pyramidal Matching és a Képmegértés két másik módszer az eljárás sebességének növelésére.

Scale- and Rotatio-Invariant Matching (Skálázás- és forgatás invariáns illesztés)

A Normalizált Kereszt-korreláció addig használható, míg a minta nem skálázott vagy nincs kb.5-10 foknál jobban elforgatva. Kiterjesztése azokra az esetekre meglehetősen bonyolult feladat. A skálázás invariáns illesztéshez az eljárás újbóli ismételése szükséges a „template” átméretezése után. Ez hatalmas mennyiségű plusz számítást jelent az eljárás során. A forgatás normalizálása pedig még ennél is bonyolultabb. Ha van valami információnk az elfordulás mértékéről, úgy elforgatva a „template”-et, könnyedén elvégezhető a korreláció. Ha az elfordulásról nem tudunk semmit, akkor a „template” teljes forgatása szükséges az illesztés sikeréhez.

Pyramidal Matching (Piramis módszer)

Kisebb területen gyorsabban el lehet végezni a mintaillesztést. A Piramis módszer mind a kép, mind a minta térbeli méretét lecsökkenti. Ez a mérték lehet például az eredeti értékek egynegyede. A mintaillesztés először a kicsinyített képeken végezzük el. Így jelentősen megnöveljük a keresés sebességét. A művelet elvégzése után már tudjuk, hogy az eredeti méretű kép melyik tartományában kell a mintát keresni.

Image Understanding (Képmegértés)

A „pattern matching features” a pixeleknek egy szembeötlő mintázata, amik meghatároznak egy „template”-et. Mivel a legtöbb kép tartalmaz redundáns információkat, a képen található összes információ használata a mintaillesztéshez időigényes lenne. Az IMAQ Vision nem egységes mintavétel technikát használ, ami magába foglalja a képmegértést így teljesen és hatékonyan leírja a „template”-et. Ez az intelligens mintaillesztő technika speciálisan tartalmazza a szegély- és terület-pixelek kombinációját. Az IMAQ Vision hasonló technikát használ, mikor a felhasználó jelzi, hogy a minta feltehetően elforgatva található a képen. Ez a technika speciálisan kiválasztott templatepixeleket használ, amik értékei - vagy viszonylagos

változások az értékekben - kifejezi a minta forgatását. A „template” intelligens illesztése csökkenti a redundáns információkat, és kiemeli a jellegzetességeket, így téve hatékonyabbá a kereszt-korrelációt. Az IMAQ Vision mintaillesztése alkalmas olyan objektumok helyeinek meghatározására, melyeknek különbözik a mérete ($\pm 5\%$) és orientációja (0 és 360 fok között).^[16]

3.5.2. Példa korrelációs mintaillesztésre:



7. ábra: kép, minta, és a kettő korrelációjának eredménye

A képen keressük a mintát korreláció segítségével. Az eredményként kapott képen minél nagyobb egy érték, annál pontosabb az illeszkedés az adott pontban.

3.6. Diszkrét koszinusz transzformáció

A diszkrét koszinusz transzformáció (DCT) a gyors Fourier transzformáció egy speciális fajtája. Abban különbözik a diszkrét Fourier-transzformációtól, hogy komplex számok helyett csak valósakkal dolgozik. A következő képlettel számolható:

$$y(k) = \sqrt{\frac{2}{N}} \alpha(k) \sum_{n=0}^{N-1} x(n) \cos\left(\frac{(2n+1)k\pi}{2N}\right)$$

ahol:

$$- \alpha(k) = \begin{cases} \frac{1}{\sqrt{2}} & \text{ha } k = 0 \\ 1 & \text{ha } k \neq 0 \end{cases}$$

- $x(n)$ az X bemeneti számsor n -edik eleme

- $y(k)$ DCT{X} k -edik eleme.

Inverze:

$$x(n) = \sqrt{\frac{2}{N}} \sum_{k=0}^{N-1} \alpha(k) y(k) \cos\left(\frac{(2n+1)k\pi}{2N}\right)$$

ahol:

$$\alpha(k) = \begin{cases} \frac{1}{\sqrt{2}} & \text{ha } k = 0 \\ 1 & \text{ha } k \neq 0 \end{cases}$$

- N a bemeneti szekvencia DCT{X} hossza
- x(n) az X kimeneti számsor n-edik eleme
- y(k) DCT{X} k-adik eleme.

A diszkrét koszinusz transzformáció Kétdimenziós esetben az alábbi formában írható fel:

$$y(u, v) = \sqrt{\frac{2}{M}} \sqrt{\frac{2}{N}} \alpha(u) \alpha(v) \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} x(m, n) \cos\left(\frac{(2m+1)u\pi}{2M}\right) \cos\left(\frac{(2n+1)v\pi}{2N}\right)$$

ahol :

- M és N a kétdimenziós tömb sorainak és oszlopainak a száma
- x(m,n) az X m-edik sorában és n-edik oszlopában található elem
- y(u,v) DCT{X} u-adik sorában és v-edik oszlopában található elem

$$\alpha(u) = \begin{cases} \frac{1}{\sqrt{2}} & \text{ha } u = 0 \\ 1 & \text{ha } u \neq 0 \end{cases}$$

$$\alpha(v) = \begin{cases} \frac{1}{\sqrt{2}} & \text{ha } v = 0 \\ 1 & \text{ha } v \neq 0 \end{cases}$$

Inverze:

$$x(m, n) = \sqrt{\frac{2}{M}} \sqrt{\frac{2}{N}} \sum_{u=0}^{M-1} \sum_{v=0}^{N-1} \alpha(u) \alpha(v) y(u, v) \cos\left(\frac{(2m+1)u\pi}{2M}\right) \cos\left(\frac{(2n+1)v\pi}{2N}\right)$$

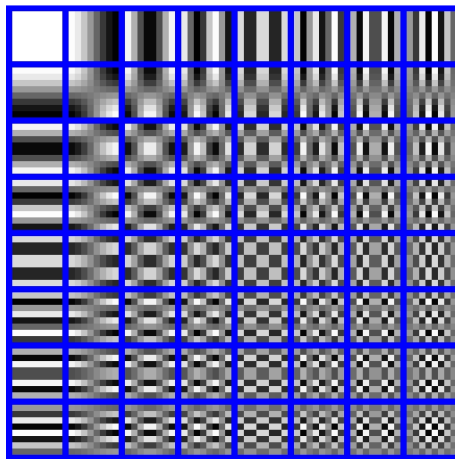
ahol :

- M és N a kétdimenziós $DCT\{X\}$ tömb sorainak és oszlopainak a száma
- $x(m,n)$ az X kimeneti mátrix m-edik sorában és n-edik oszlopában található elem
- $y(u,v)$ a bemeneti $DCT\{X\}$ u-adik sorában és v-edik oszlopában található elem

$$\alpha(u) = \begin{cases} \frac{1}{\sqrt{2}} & \text{ha } u = 0 \\ 1 & \text{ha } u \neq 0 \end{cases}$$

$$\alpha(v) = \begin{cases} \frac{1}{\sqrt{2}} & \text{ha } v = 0 \\ 1 & \text{ha } v \neq 0 \end{cases}$$

A $\cos\left(\frac{(2m+1)u\pi}{2M}\right) \cos\left(\frac{(2n+1)v\pi}{2N}\right)$ tagot bázisfüggvénynek nevezzük, amit 8x8-as blokk esetén az alábbi ábrával szoktak szemléltetni.

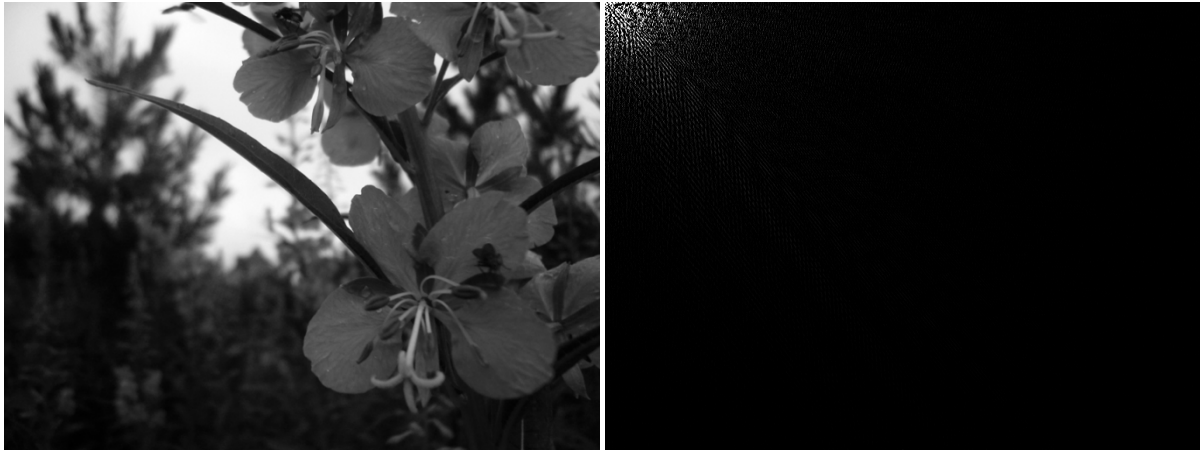


8. ábra: DCT bázisfüggvénye 8X8-as blokk esetén

A jobb felső sarokból indulva balra és lefelé haladva növekszik a frekvencia.

A transzformációt elvégezve a kapott mátrixnak a (0,0) pontja lesz a blokk átlagos értéke. E körül helyezkednek el a meghatározó alacsonyfrekvenciás elemek. Távolodva a részleteket tartalmazó magas frekvenciájú elemek vannak. Ebből az inverz transzformáció segítségével az eredeti kép veszteségmentesen visszaállítható.

A DCT-t elsősorban JPEG tömörítéshez használják. A képfeldolgozásban szűrőként alkalmazhatjuk. A transzformációt követően a blokk első elemét kinullázzuk, majd az inverz transzformáció elvégzése után megkapjuk a szűrt képet. Hatására a sötét területek gyenge változásai eltűnnek.



9. ábra: Kép és transzformáltja

A LabVIEW rendelkezik olyan beépített VI-jal, mely egy bemeneti X számsorra végrehajtja a DCT-t, a fentebb említett képlet alapján. 2 dimenziós esetben a következő képen jár el:

- A kétdimenziós X szekvencián sorról sorra haladva elvégzi az 1D-DCT-t. A kimenetet Y' -vel jelöli.
- Elvégzi az 1D-DCT-t Y' -n, oszlopról oszlopra haladva. Az eredmény már maga lesz a $DCT\{X\}$.

A kétdimenziós Inverz DCT-t is hasonló módon számolja:

- Végrehajtja az egydimenziós Inverz DCT-t sorról-sorra haladva a $DCT\{X\}$ bemeneti mátrixon. A kimenetet Y' -vel jelöli.
- Elvégzi az egydimenziós Inverz DCT-t oszlopról-oszlopra haladva az Y' -n. A művelet eredménye most már maga az X mátrix.^{[18] [19] [20] [21]}

4. A programról

A program egy referencia alapú képösszehasonlításos vizsgálatot végez. Mivel a LabVIEW kifejezetten vizsgálati rendszerek fejlesztésére készült, így ez volt a legalkalmasabb a program elkészítésére. Ugyanakkor rengeteg beépített VI áll rendelkezésre, melyek a képfeldolgozást segítik.

4.1. LabVIEW

Laboratory Virtual Instrumentation Engineering Workbench

Hivatalos honlap: <http://www.ni.com/labview/>

A LabVIEW egy grafikus programozási környezet. A National Instruments mérési-, vizsgálati- és ellenőrzési rendszerek fejlesztésére dolgozta ki. Segítségével virtuális műszereket (Virtual Instrument - VI) hozhatunk létre. A VI két részből tevődik össze. Tartozik hozzá egy Frontpanel és egy Block Diagram. A Frontpanelen található elemeket két csoportra oszthatjuk. Control-nak (vezérlőnek) azokat az elemeket nevezzük, amikkel értéket állíthatunk be, illetve a program működését befolyásolhatjuk, pl. gombok. Az Indicator (kijelző) csoportba pedig az értéket megjelenítő elemeket soroljuk, pl. grafikonok. A Block Diagram az adatfolyam grafikus megjelenése, ami a tényleges programkódot tartalmazza. A program csomópontokból és vezetékekből épül fel, ami párhuzamosan és több szálon futtatható. A csomópontok közé tartoznak a beépített függvények, az alprogramok (SubVI-ok) és a struktúrák. Ezeket köthetjük össze a huzalokkal, így alakítva ki az adatfolyamot.

A Case struktúrát elágazásnak is szokták nevezni. Az esetválasztójára kapott értéknek megfelelő ágában folytatódik a program futása. A struktúrák között találhatóak a ciklusok is, amik egy programrészlet többszöri ismétlését teszik lehetővé. A While ismétlését egy logikai érték („igaz”/”hamis”) szabályozza. A For-ciklusnak előre meg kell adni az ismétlések számát. A ciklusokhoz adhatunk egy úgynevezett Shift-Register-t, ami arra szolgál, hogy egy értéket a ciklus következő lefutásakor is fel tudjunk használni. Struktúrák alkalmazása közben felmerülhet a probléma, hogy egy elemhez a program különböző részeiben hozzá szeretnénk férni, de annak csak egyetlen ikonja van. Erre ad megoldást a lokális változó, amely az adott elemre történő hivatkozás. Előnye hogy segítségével egy elem értékét beállíthatjuk vagy lekérdezhetjük, függetlenül attól, hogy az a Frontpanelen eredetileg indikátor vagy kontrol

volt. Az Application Control-ok közt található a Property Node (tulajdonság csomópont), ami az egyes elemek tulajdonságainak a kezelésében nyújt segítséget. Egyszerre több tulajdonságot is kezelhetünk, amiknek az értékeit a lokális változókhoz hasonló módon kezelhetjük.^[22]

4.2. A program alapjai

A program alapjául szolgáló ötletet két tanulmány adta. Az egyik a „Wavelet-Based Printed Circuit Board Inspection System”^[23], a másik a „Design of automatic vision-based inspection system for solder joint segmentation”.^[24]

Az első jegyzet a referencia alapú vizsgálatok közül a képösszehasonlításra alapuló módszert részletezi. Leírja, hogy az összehasonlítást két részre kell bontani. Először egy pozitív, majd egy negatív képet készítünk. A pozitív úgy jön létre, hogy a referenciából kivonjuk a tesztet. Ekkor az utóbbiról hiányzó elemek jelennek meg az eredményen. A negatív, pont ennek az ellenkezője. A tesztképből vonjuk ki a referenciát, ezáltal megkapva azokat az objektumokat, melyeknek nem lenne szabad jelen lenniük. A két kép még tartalmaz zajokból adódó nem kívánt információkat, valamint nincs egységes háttérük sem. Ezeket egy morfológiai eljárásokhoz hasonló elven működő szűrővel eltávolítja. Végül a két képet egyesítve megkapjuk az összes jelentős eltérést. A leírtak alapján a program is elvégzi a képkivonásokat, de már előkészített, bináris képeken. A zavaró részletek eltüntetésére viszont nem a jegyzetben megadott szűrőt alkalmazza.

A másik tanulmány a forrasztások helyének meghatározását tárgyalja. Elsőként a diszkrét koszinusz transzformációt részletezi, ami segítségével a háttér gyenge változásai eltüntethetőek, és kiemelhetőek a világos részeket. Az eljárás valóban hasznosnak bizonyult. Következőnek azt mutatja be, hogy az RGB színrendszerből hogyan lehet áttérni YIQ-ba. Lényeges eltérés a szürkeárnyaltos képhez képest nem látható, ráadásul a művelet elvégzése rendkívül időigényes feladat. Egy automatikus küszöbérték meghatározásának ismertetése következik, melynek számolási módja a pixelek értékeinek átlagolásán alapul. Ennek ismerete hasznos lehet más esetekben, de ide nem illik. Végül ír egy objektumon található lyuk betöltéséről, ami egyébként morfológiai eljárásokkal is megvalósítható.

4.3. A vizsgálat menete

A vizsgálat kép összehasonlításán alapul. A referencia és a tesztelni kívánt panelekről felvételt készítünk. Megnyitjuk a referencia és a teszt képet. A kamera torzításából adódó hibákat kalibrációval korrigálhatjuk. Ehhez szükség van egy rácsot (grid-et) tartalmazó képre. Kalibráció során a rács alapján számolja a program a torzulás mértékét, majd az eredménytől függően javítja a képet. Ha referencia és a tesztkép a felvételen más-más pozícióban helyezkedik el, az összehasonlítást nem lehet elvégezni. Ebben az esetben a tesztképet a referenciához igazítjuk. A képek éleihez koordináta-rendszert illesztünk. A koordináta-rendszerek kiindulási pontjai megadják az eltolást x és y irányban, a szögük pedig az két kép elfordulását egymáshoz képest. Ezen adatok alapján a tesztképet már pozícionálni tudjuk. Következő lépésként a képeket részekre osztjuk. Erre azért van szükség, mert kisebb területen gyorsabban elvégezhetőek a képfeldolgozási műveletek, másrészt ez is segíti a pontosabb illeszkedést. A tesztdarabokat a referenciadarabok alapján keressük mintaillesztés segítségével. Tulajdonképpen a darabokat fogjuk egymáshoz viszonyítani, és nem az egész képet.

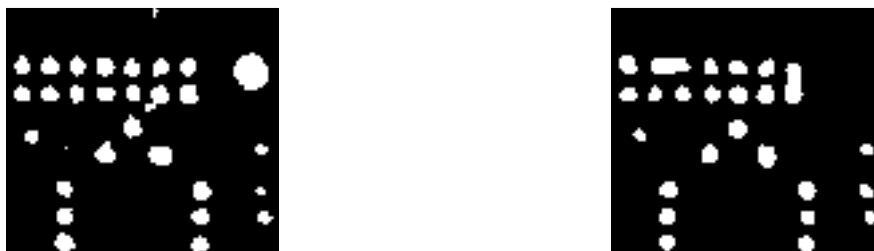


9. ábra: referencia és tesztkép részletei

A forrasztások kiemelése érdekében elvégezzünk egy Diszkrét Koszinusz Transzformáción alapuló szűrést, majd egy alacsony küszöbérték mellett binarizáljuk a képet. A bináris képen már elvégezhető az összehasonlítás.



10. ábra: referencia és teszt darab a DCT elvégzése után



11. ábra: binarizált referencia és teszt darabok

A referenciából kivonva a tesztet egy pozitív képet kapunk, a tesztből kivonva a referenciát pedig egy negatívot. A keletkezett képek még tartalmazhatnak zajokat, nem kívánt információkat, melynek oka lehet pl. a még mindig nem tökéletes fedés, vagy egyéb fényvisszaverődésből adódó „zajok”.



12. ábra: pozitív (referencia - teszt) és negatív (teszt - referencia) képek

Elkészítjük a pozitív és negatív kép részecskék váz-szerkezetét (skelton), melyet kivonunk a megfelelő pozitív és negatív képből. Ezáltal eltűnnek a vékony objektumok, a szélesebbek pedig szétesnek.



13. ábra: objektumok váz-szerkezete (skelton) a pozitív és negatív képeken



14. ábra: objektumok a skelton kivonása után

Hogy ezt megszüntessük, egy morfológiai bezárást kell végrehajtani.



15. ábra: objektumok a morfológiai bezárás után

Az esetlegesen még mindig jelenlévő 1-2 pixelnyi zajokat részecskeszűréssel eltávolítjuk. Ami a képen marad, azt kell hibának tekinteni.



16. ábra: részecskeszűrés után a tényleges hibák: többletek és hiányok

A program ezeket összegyűjti, a kijelzőn megjeleníti és egy fájlba menti.

4.4. A program kezelése

A program kezelőfelületén található egy kijelző és egy kontrolpanel. Az utóbbin végezhetjük el a beállításokat és a program vezérlését.

4.4.1. Vizsgálat előkészítése

Egy vizsgálat előkészítéséhez először is adjuk meg a referenciakép elérési útvonalát. Amennyiben kalibrálásra szorul, adjuk meg a rácsot tartalmazó képet a „Grid”mezőben, és kapcsoljuk be a „Kalibral” gombot. Ezek után a „Megnyit” gombra kattintva a referenciakép megjelenik a kijelzőn. A „DarabHelyLement” mezőbe annak a fájlnak az elérési útvonalát adjuk meg, ahova a vizsgálni kívánt területek helyeit akarjuk menteni. A kijelző baloldalán található ROI eszközök közül válasszuk a négyszögletest, és jelöljük ki egy vizsgálni kívánt tartományt, majd nyomjuk meg a „Darabol” gombot. Ezzel a módszerrel adjuk meg az összes olyan területet, amin a vizsgálatot el akarjuk végezni.

4.4.2. Beállítások

A beállításokkal a vizsgálat során történő kiértékelést befolyásolhatjuk. Tökéletes képek esetén csak annyit kell megadnunk, hogy a hibás területeket milyen színnel jelezze. A „Skelton” az eredményül kapott kép objektumainak elkészíti a váz-szerkezetét, majd azt kivonja belőle. A „Morfologia” egy bezárást hajt végre a képen. A „ParticleFilter” pedig eltávolítja („Remove Particles”) a „Selection Values” által megadott részecskéket. A „Beallitasok1” alatt a forraszhiányokra, a „Beallitasok2” alatt pedig a forrasztöbbletekre vonatkozó feltételek találhatók.

4.4.3. Vezérlés

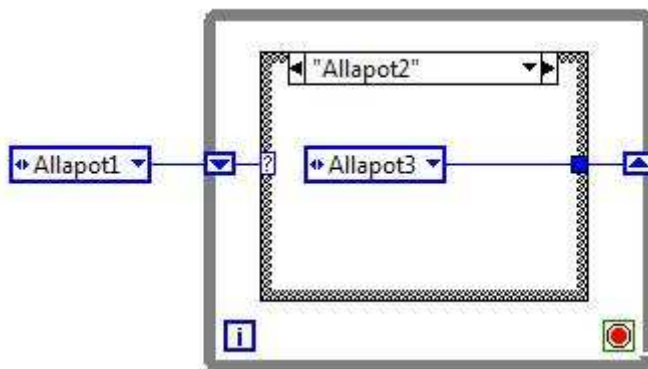
A „DarabHelyBetolt” mezőn adjuk meg annak a fájlnek az elérési útját, amelyikben a vizsgálandó képrészek helyei vannak lementve. Az „Eredmeny helye” mezőn megadott fájlba fog a vizsgálat eredménye kerülni. A referenciakép betöltésére itt is a korábban leírtak érvényesek. A vizsgálandó képeket sorban a „Tesztek” közé vegyük fel. Amennyiben ezek a referenciához képest elmozdulva találhatók, nyomjuk be a „Helyretesz” gombot, hogy a program majd elvégezze az igazításukat. A „kesleltetes” két állapot közti átmenet idejét lassítja a megadott értéket ezredmásodpercben értve. A vizsgálatot a „START” gomb lenyomásával indíthatjuk. Az „Allapot” rész segítségével nyomon követhetjük, hogy a vizsgálat éppen hol tart. Egy feladat elvégzése után világít a „KESZ” jelzés. A programot a „STOP” gomb megnyomásával állíthatjuk le. A „Leallit” gomb hatására a program befejezi működését mihelyt elkészült az aktuális művelettel.

4.5. A program szerkezete

4.5.1. Állapotgép

A program vezérlését egy állapotgépen alapuló módszer segítségével oldottam meg. A jelenlegi állapot és a környezet együttesen határozzák meg, hogy a programnak mi lesz az új állapota. LabVIEW-ban állapotgép létrehozásához egy While-ciklusra, egy Case-struktúrára és egy Enumra (felsorolásos típusra) van szükség. Az Enum-ban felvesszük a különböző állapotok megnevezését. A Case-struktúrát a ciklusban helyezzük el, ezzel biztosítva a program folyamatos működését. A Case esetválasztójába bekötjük az állapotneveket tartalmazó Enumot, és minden eleméhez hozzárendelünk egy esetet, így hozva létre az

állapotokat. Egy állapotban szerepelni kell az őt követő állapotra történő utalásnak, amit szintén Enummal adunk meg. Azt bekötjük a ciklus Shift Registerébe, ezzel biztosítva, hogy a következő lefutáskor már az új állapotba lépjünk.



17. ábra: állapotgép megvalósítása

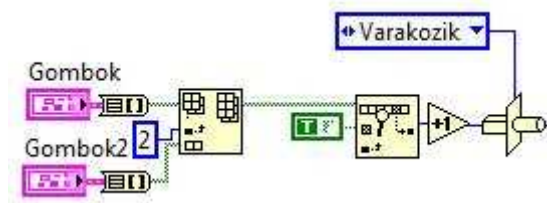
A programban a következő állapotok szerepelnek: „Várok”, „Elkészít”, „Stop”, „Darabol”, „Feldolgoz”, és „Megnyit”.

4.5.1.1. Várok

Van egy állapot, ami a program szempontjából kitüntetett szereppel bír, ez a „Várok”. Induláskor megtörténik a különböző elemek inicializálása, majd utána egyből ez az állapot következik. Várok alatt azt kell érteni, hogy a program addig nem hajt végre különösebb műveletet, amíg arra a felhasználótól utasítást nem kap. A „Várok” állapot után ismét várok következik, teszi ezt a program mindaddig, amíg gombnyomás nem történik. Gombnyomás hatására az Enum egy másik eleme kerül a Shift Registerbe, és a ciklus következő lefutásánál a kiválasztott állapotba térünk át. Ekkor elkezdődik az érdemi munka, majd annak végeztével ismét a várok következik.

Az összetartozó gombokat a Frontpanelen Cluster segítségével csoportba foglaljuk, ezzel is kiemelve azokat. A gombok értéke csak „igaz”(true) vagy „hamis”(false) lehet. A Clusterekből létrehozunk egyetlen logikai (boolean) típusú tömböt. Egy másik módszer ennek a tömbnek a létrehozására, hogy közvetlenül a gombokból építjük azt fel a Build Array segítségével. Ebben a tömbben elvégzünk egy keresést. Ha találunk benne true értéket, az azt jelenti, hogy történt gombnyomás, és egyúttal megkapjuk a megnyomott gomb sorszámát. Ha a felhasználó nem nyomta meg egyik gombot sem, a keresés eredménytelen és a „-1” értékkel tér vissza. Mivel az indexelés 0-tól indul, ezért a lehetséges „-1” miatt az eredményt eggyel

növelni kell. Az Enumból kiválasztjuk az így kapott számnak megfelelő indexű elemet. Ezt a műveletet nevezzük kasztolásnak. A kapott elem mondja meg, hogy mi lesz a programnak a következő állapota. Arra kiemelten oda kell figyelni, hogy a gombok és a hozzájuk rendelni kívánt állapotnevek sorszámai megegyezzenek, különben egészen mást fog csinálni a program, mint amit szeretnénk.



18. ábra: állapot kiválasztása

4.5.1.2. Stop

Ebben az állapotban tényleges számítás nem történik, mégis igen fontos része a programnak. A While-ciklus akkor fejezi be az ismétlést, amikor a Loop Condition (ismétlési feltétel) „igaz” értéket kap. A „Stop” állapotban fejeztetjük be a ciklus ismétlődését, e nélkül a végtelenségig futna. Itt lehetne még különböző műveleteket végezni, mint például fájllezárás vagy memóriaterület felszabadítás, de azokat célszerű a cikluson kívül elhelyezni.

4.5.1.3. Megnyitás

A „Megnyit” állapot során csupán a referencia képet tölti be a program. Amennyiben szükség van annak kalibrálására, úgy a „Grid”-et tartalmazó kép is megnyitásra kerül. A művelet elvégzése után a kalibrált kép referenciaként kerül továbbvitelre. Végezetül a várakozó állapotba való áttérés történik, melyben a felhasználó megadja, hogy mit akar a most betöltött képpel kezdeni.

4.5.1.4. Darabolás

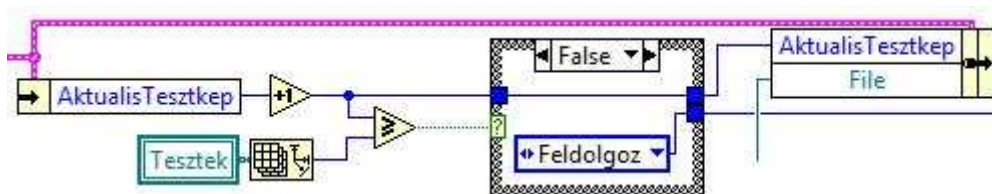
A korábban megnyitott kép részekre osztását végezzük a „Darabol” állapotban. A felhasználó által kijelölt területet a „Darabok” közé illeszti a program. Eközben ezeknek a részeknek a helyét egy fájlba menti egy későbbi használat érdekében. A darabolás után ismét várakozó állapot következik, onnan újabb darabot hozhatunk létre, vagy továbbléphetünk.

4.5.1.5. Előkészítés

Az „Előkészít” állapotban a referenciakép megnyitása, és feldolgozása történik meg a későbbi használat megkönnyítése érdekében. Egyúttal itt esedékes annak a fájlnak a létrehozása, amibe a vizsgálat eredménye fog kerülni. Az előkészítés után a program már nem a várakozó állapotba kerül vissza, hanem a feldolgozásba lép át.

4.5.1.6. Feldolgozás

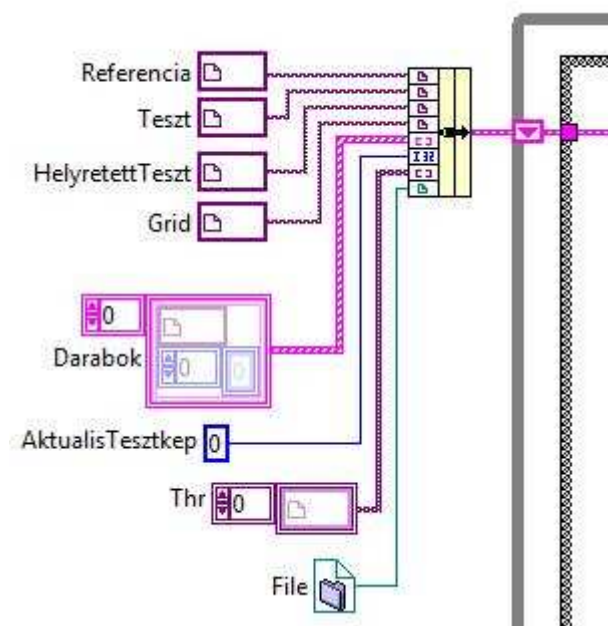
A „Feldolgoz” állapot annyiban különleges, hogy ide nem lehet közvetlenül a várakozásból eljutni, csak az előkészítés után kerül rá a sor. A tesztképet a megnyitása után ugyanúgy elő kell készíteni, mint korábban a referenciát. Mikor ez megtörtént, következhet a program érdemi része, a kiértékelés. A feldolgozás rész még egy érdekessége, hogy az azt követő állapot függ a környezettől is. Amennyiben van még vizsgálandó kép, úgy újra a „Feldolgoz” következik, ellenkező esetben a program visszatér a várakozásba.



19. ábra: állapotváltás a feldolgozás végén

4.5.2. Cluster használat

A programban szereplő különböző elemek az egyes állapotokban más-más értéket vehetnek fel, amikre a későbbiekben szükség lehet. Ahhoz, hogy ezekhez az értékekhez egy másik állapotban is elérhetőek legyenek, át kell oda vinni őket a While-ciklus Shift Registere segítségével. Ez persze rengeteg huzallal járna, ami a programkódot átláthatatlanná tenné. Hogy elkerüljük ezt az esetet, célszerű az elemeket inkább csoportba foglalni egy Cluster segítségével. Programozás technikai szempontból hasznosnak bizonyult ez a módszer, mivel így csupán egyetlen huzal alkalmazására van szükség. Egy elem értékének eléréséhez elegendő csak az elem leválasztása a csoportból. Az Unbundle By Name segítségével a leválasztás könnyedén megtehető. Amennyiben változtatni szeretnénk, rendelkezésre áll a Bundle By Name, melyben kiválasztjuk a módosítandó elemet, és a bemenetére az új értéket kötjük.



20. ábra: elemek egyetlen csoportba történő foglalása

4.6.A program részletezése

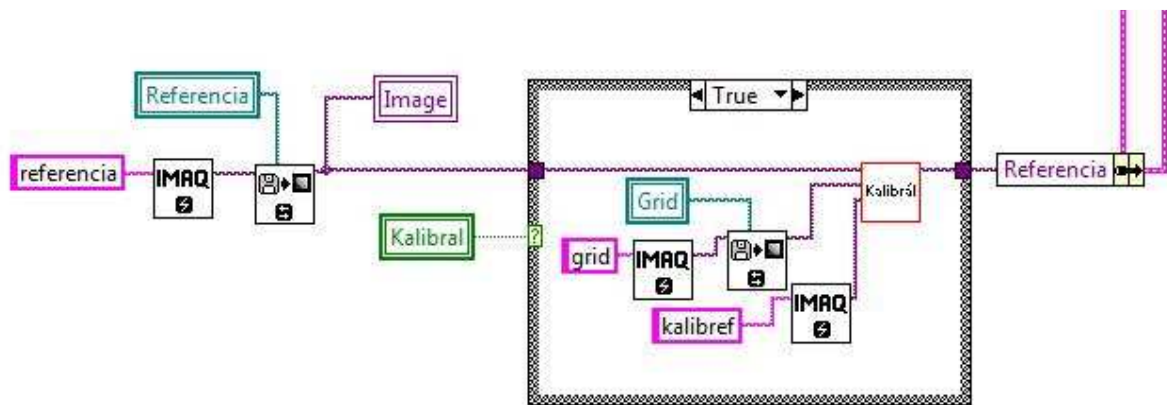
4.6.1. Megnyitás

Megnyitás során a referenciaképet töltjük be, szükség esetén kalibráljuk is, majd a Cluster „Referencia” nevű elemébe tesszük.

Legelső lépésként a képnek egy ideiglenes tárhelyet kell foglalni a memóriában, amit az „IMAQ Create” VI tesz meg. Paraméternek meg kell adni egy nevet („Image Name”), és esetleg a kép típusát attól függően, hogy szürkeárnyaltos (Grayscale U8) vagy színes (RGB U32) képet fogunk kezelni.

Ezt követően az „IMAQ ReadFile” VI beolvassa a „File Path”-ra adott elérési út által hivatkozott képet, és elhelyezi azt az „Image” bemeneten kapott memóriaterületre.

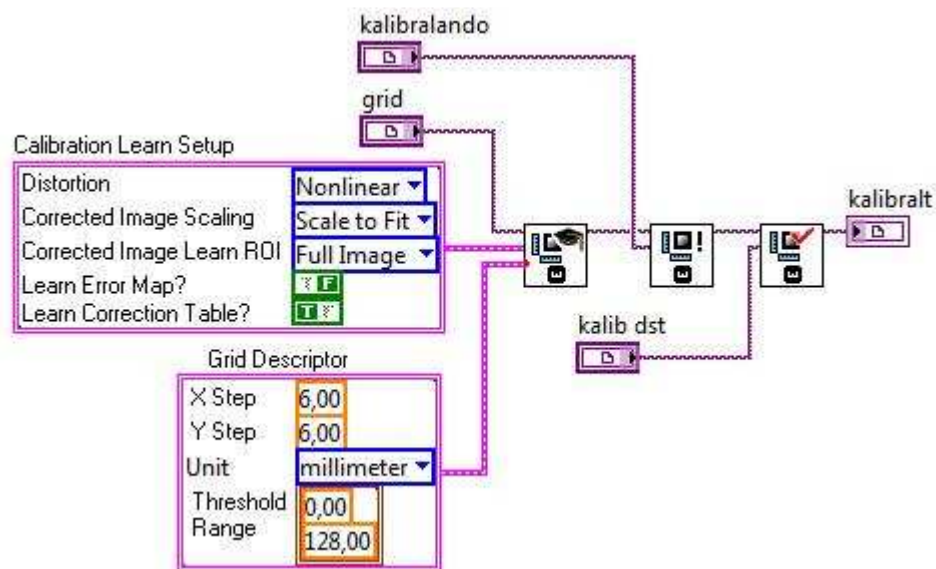
Amennyiben a Frontpanel „Kalibrál” nevezetű gombja „igaz” értékre van állítva, egy Case-struktúra true ágában megtörténik a kalibráció. Ehhez azonban szükség van egy úgynevezett Grid-et tartalmazó képe, amit az előbbieken tárgyalt módon be kell olvasni. Hogyha nem akarjuk kalibrálni, akkor a false ágban egyszerűen csak továbbvisszük a beolvasott képet.



21. ábra: Megnyitás

4.6.1.1. Kalibrálás

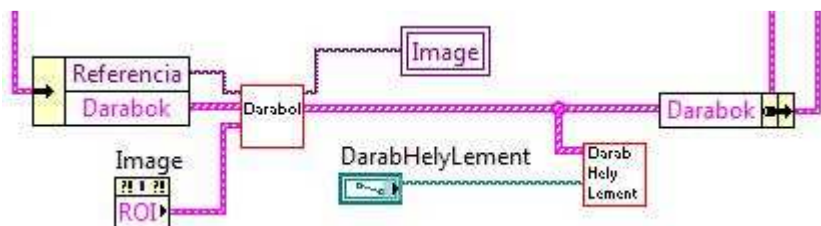
Kalibrálással a kamera torzulásából adódó hibákat lehet korrigálni. A VI három bemenettel rendelkezik. A „kalibrando” képen elvégezzük a javítást, majd a „kalib dst”-be mentjük. A „grid” a kalibrációhoz szükséges rácsot tartalmazza. Első lépésként az „IMAQ Learn Calibration Template” megkapja a rácsot, és annak leírását („Grid Descriptor”). A „Calibration Learn Setup” a művelet beállításait tartalmazza. Második lépésben az „IMAQ Set Calibration Info” a „grid” kalibrációs információit hozzárendeli a javítani kívánt képhez. Végül az „IMAQ Correct Calibrated Image” elvégzi a javítást.



22. ábra: Kalibrál VI

4.6.2. Darabolás

Darabolás során a referencia kép azon részeit adjuk meg, amik a vizsgálathoz szükségesek. A Clusterből a „Referencia” és „Darabok” elemekre van szükség. Az „Image Display” tulajdonságai közül lekérdezzük a ROI-t. A „Darabol” VI a ROI alapján meghatározza az aktuális részt, és azt beteszi a „Darabok” közé, amiket a „DarabHelyLement” VI egy fájlba ír ki.



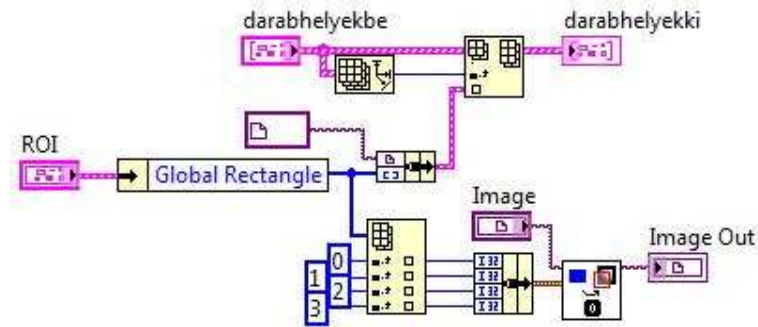
23. ábra: darabolás

4.6.2.1. Darabol VI

A „Darabol” VI végzi a darabolás érdemi részét. Paraméterként megkapja a referenciaképet, a felhasználó által megadott ROI-t, és a Darabokat tartalmazó tömböt. Egy ilyen Darab két részből fog állni. Az egyik része egy kép, amin a referenciának egy adott területe van kivágva. A másik része egy „Global Rectangle”, azt mondja meg, hogy a Darab hol található a nagy képen.

A ROI-nak a „Global Rectangle” nevű eleme alatt azt a négyzetet kell érteni, ami a képen kijelölt tartományt teljesen magába foglalja. Ezt, és egy üres kép konstans csoportba foglalunk, ezáltal létrehozva egy újabb Darabot, amit aztán a már meglévőket tartalmazó tömbbe teszünk.

A „Darabol” VI mindeközben a referencia képre ráhelyez egy Overlay-t. A művelet a program működésére ugyan nincs hatással, inkább csak a felhasználó munkáját segíti. Az Overlay-ek a már létrehozott Darabokat mutatják, ezzel biztosítva, hogy kétszer ne szerepeljen ugyanaz a rész, és az átfedések is minimálisak legyenek. A megjelenítést az „IMAQ Overlay Rectangle” végzi, de ennek nem felel meg a „Global Rectangle”, így egy hagyományos „Rectangle”-t kell készíteni. A „Global Rectangle” első négy elemét lekérdezzük, és azokat csoportba foglaljuk.

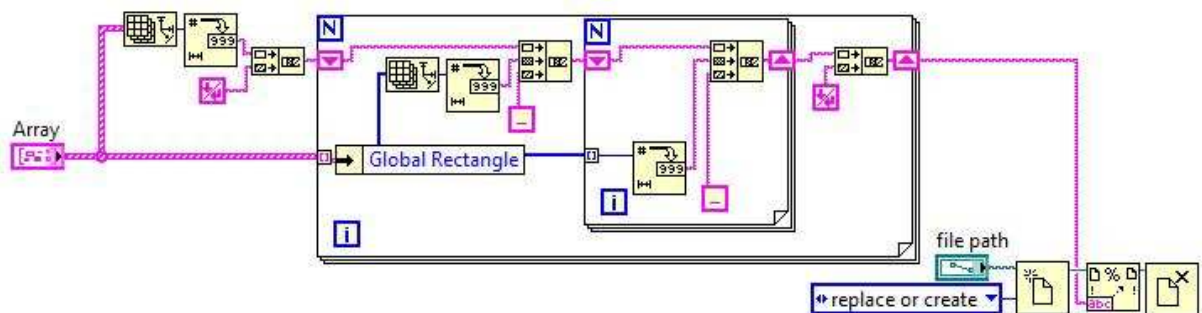


24. ábra: Darabol VI

4.6.2.2. DarabHelyLement VI

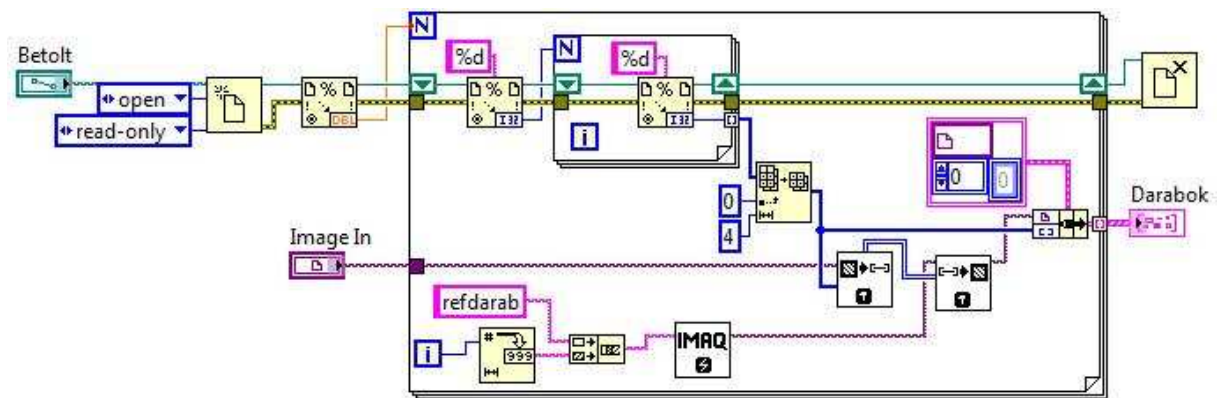
A VI paraméterként megkapja a Darabokat tartalmazó tömböt, és annak a fájlnek ez elérési útját, amibe azt menteni szeretnénk. Az adatokat egyetlen sztringbe gyűjtjük össze, ezáltal a fájlba történő írás is egyetlen lépésben megvalósítható. Array Size segítségével meghatározzuk a tömb méretét, majd a kapott értékből Number To Decimal String használatával sztringet konvertálunk. A tömbből ehhez fogjuk hozzáfűzni a Darabok adatait. Az adatok kinyeréséhez két For-ciklusra van szükség. A külsővel a tömb elemeit vesszük végig. Leválasztjuk „Global Rectangle”-t és meghatározzuk az elemeinek a számát az előbbiekben kifejtett módon. Concatenate Strings használatával bővítjük a már meglévő sztringet. A második For-ciklussal a „Global Rectangle” elemeit írjuk az eddig kinyert adatok közé. Mindezek közben Space Contant és End of Line Constant használatával tagoljuk a sztringet, hogy átláthatóbb legyen.

Az „Open/Create/Replace File”-nak két bemenete lényeges. Az egyik a „file path”, oda azt a fájllelési utat kötjük, amit a „DarabHelyLement” VI is kapott. A másik paramétere az „Operation”. Itt azt mondjuk meg, hogy milyen műveletet szeretnénk végrehajtani. Jelen esetben ez „replace or create”. Amennyiben a fájl létezik, úgy azt felülírja, ellenkező esetben pedig létrehozza. A tényleges írási műveletet a Format Into File végzi. Ennek végeztével a lezárás a Close File segítségével történik.



25. ábra: DarabHelyLement VI

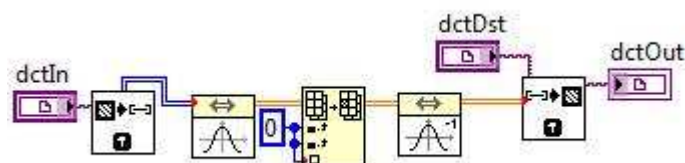
Az értékek kinyeréséhez két For-ciklusra van szükség. Az első szám azt mondja meg, hogy hány elemünk lesz, a ciklus is annyiszor fog lefutni. A következő érték azt mondja meg, hogy az aktuális elem hány részből áll, ami a belső ciklus ismétléseinek a számát jelenti. A belső ciklusban nyerjük ki ténylegesen az egyes elemek tartalmát, ami nem más, mint a Darab helyzetét jelző „Global Rectangle”. Mint már szó volt róla, egy referencia Darab két részből áll. Van egy képrészlet, és egy „Global Rectangle”. Miután fájlból beolvastuk a helyét, már csak a képet kell elkészíteni. „IMAQ ImageToArray” segítségével egy 2D-s tömböt készítünk a referenciakép „Global Rectangle” által megadott területéről. Ezt az „IMAQ ArrayToImage” konvertálja vissza képpé, így megkapva a képrészletet. A szükséges memóriaterületek nevét a ciklusváltozóval sorszámozzuk. A darabokat tömbbe helyezzük és a VI kimenetére küldjük. Ezek után már csak a fájlbezárást kell elvégezni.



27. ábra: DarabHelyBetolt VI

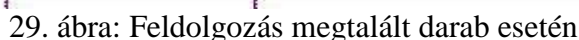
4.6.3.2. DCT VI

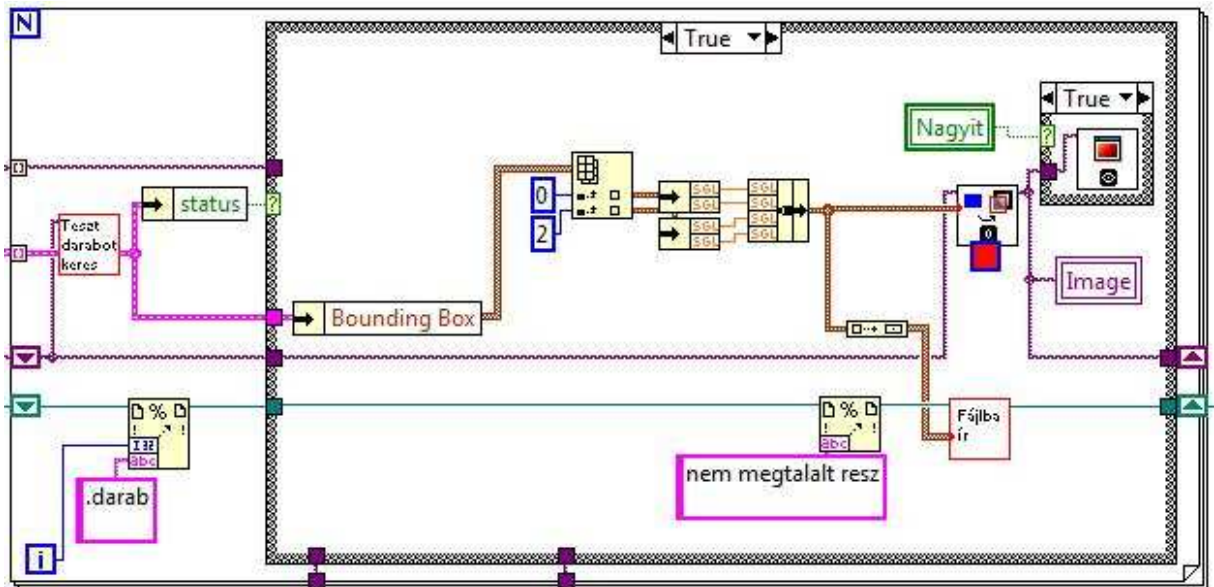
A „DCT” VI bemenetére kapott képből egy 2D-s tömböt készítünk. Ezen elvégezzük a Diszkrét koszinusz transzformációt. Az eredményül kapott mátrix kezdőpontjában (0,0) található elemet kinullázzuk. Az inverz transzformáció végrehajtása után visszakonvertáljuk képpé. Az eljárás eredményeként a sötét háttér apró változásai eltűnnek, és a fényes területek jól elkülönülnek.



28. ábra: DCT VI

A tesztképek elérési útjait tartalmazó tömbből kiválasztjuk a soron következőt, majd az alapján megtörténik a kép megnyitása. Ha szükséges, elvégezzük a kalibrálását a korábban már leírtak szerint. A kalibrálást a helyretétel követi, melynek során a referenciaképpel megegyező pozícióba helyezzük a vizsgálandó képet. Ezt követően veszünk egy referenciadarabot, és megkeressük azt a tesztképen a „TesztDarabotKeres” VI segítségével. A tesztadarabot tartalmazó csoportnak három tagja lesz. Az első a vizsgálandó képből kivágott részlet, a második a helyét meghatározó „Bounding Box”, a harmadik pedig egy logikai érték, ami azt mondja meg, hogy történt-e hiba a keresése közben. Amennyiben rendben megtaláltuk a képet, úgy elvégezzük rajta azokat a műveleteket, amit a referencia darabon is az előkészítés során: a „DCT”-t és a „Threshold”-ot. Ezek után a két képrészlet már összehasonlítható, és következhet az elemzés. Az elemzés két részből áll. Először a pozitív kivont képet, majd a negatívot készítjük el. Az eredményt fájlba írjuk, mely tartalmazza a vizsgált kép nevét, majd darabonként külön csoportosítva a hiányt és többletet. Amennyiben a tesztadarabot nem sikerült megtalálni, úgy a kijelzőre és a fájlba az eredménytelen keresés helye kerül hibaként. Az előkészületeket és a vizsgálatokat minden egyes Darab esetén elvégezzük. Az elemzést végző VI működését meghatározó értékeket a felhasználó adhatja meg a vizsgálat feltételeihez igazodva. Az eredményekkel ellátott kép teljes méretében is megtekinthető.



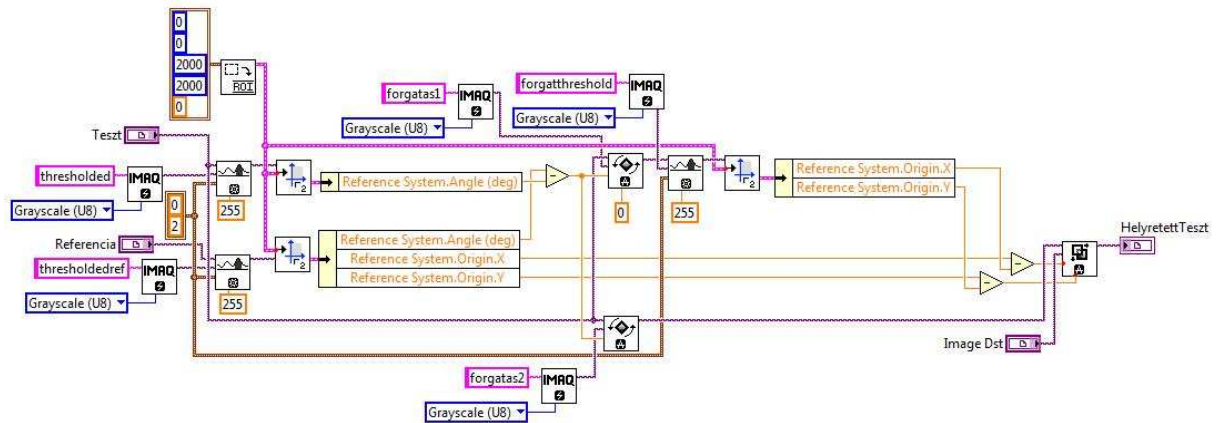


30. ábra: Feldolgozás meg nem talált darab esetén

4.6.4.1. Helyretétel

Ahhoz, hogy a vizsgálandó képet a referenciához tudjuk hasonlítani, azonos pozícióba kell vele helyezni. A VI megkapja ezt a két képet, valamint egy memóriaterületet, ahova az eredményt menti. Első lépésben mindkét képen elvégzünk egy Threshold-ot. A tartományát 0 és 2 közé állítva a háttérrel különítjük el a paneltől. Az eredmény most nem bináris kép lesz, hanem egy szürkeárnyaltos, csak 0 és 255 értékű képpontokkal.

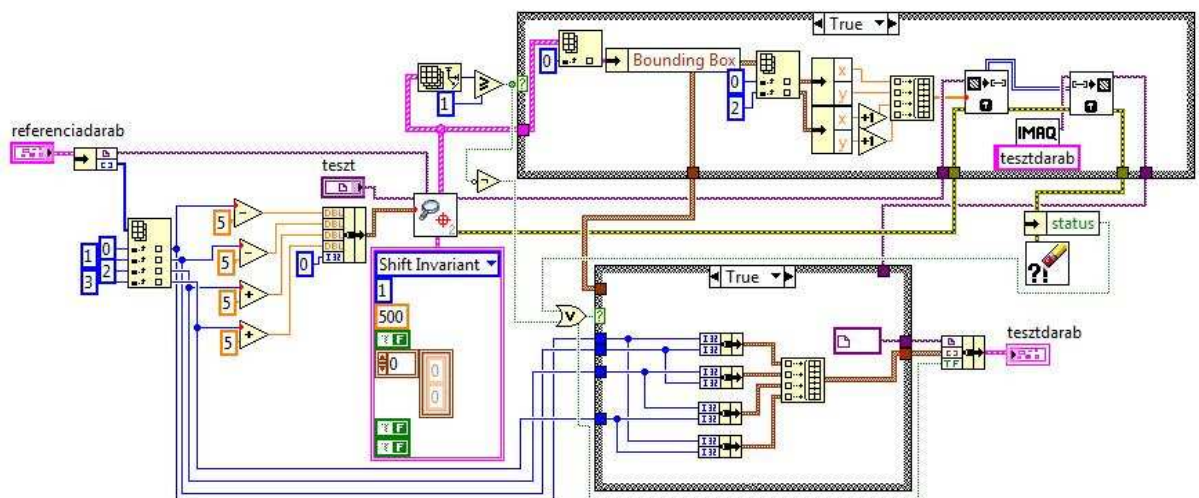
A kapott képek bal és az alsó éleihez az „IMAQ Find CoordSys (Rect) 2” segítségével egy koordináta-rendszert illesztünk. Lekérdezzük a koordináta-rendszerek dőlésszögét, és a különbségükkel megegyező szögben elforgatjuk a tesztképet. Az elforgatott képen szintén elvégzünk egy Threshold-ot, majd ismét egy koordináta-rendszer illesztés következik. Ezúttal az eredetükre vagyunk kíváncsiak. Az x koordinátákat kivonva megkapjuk, hogy vízszintesen mennyire kell eltolni a már elforgatott tesztképet, az y koordináták különbsége pedig a függőleges eltérést adja meg. Az eltolást az „IMAQ Shift” végzi.



31. ábra: Helyretesz VI

4.6.4.2. Tesztdarab keresése

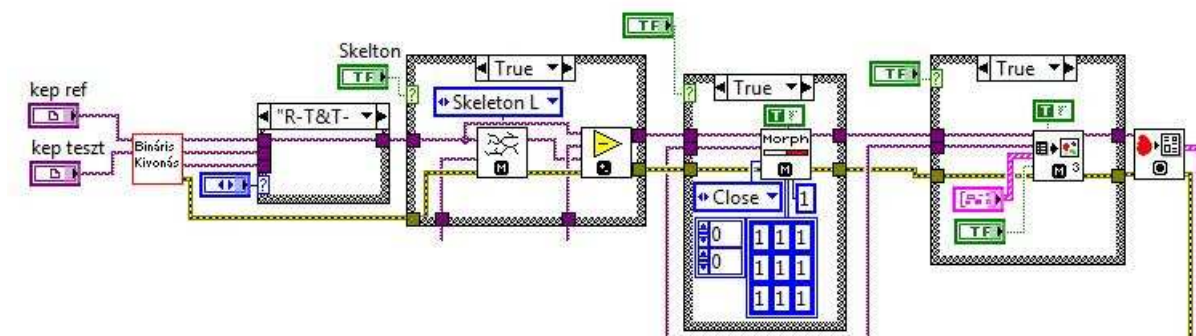
A VI a paraméterként kapott referenciadarabhoz elkészíti a tesztdarabot. Ehhez megkapja még a tesztképet. A referenciadarab „Global Rectangle” tagját minden irányban 5-5 pixellel megnöveljük, majd létrehozunk belőle egy „Search Rect”-et. A „Search Rect” határozza meg a mintaillesztés keresési tartományát. A mintaillesztést az „IMAQ Find Pattern 2” végzi, „Image”-nek a tesztképet, „Template”-nek pedig a „RefReszlet”-et adjuk meg. Mivel amúgy is csak egyetlen eredményt várunk, így a találatokat tartalmazó tömbnek az első elemét használjuk. A találat „Bounding Box” tagjából „Rectangle”-t készítünk. A tesztképnek kivágjuk a „Rectangle” által határolt területét. A tesztdarab három részből fog állni. Amennyiben a mintaillesztés sikeres volt, úgy a képet megkapja a tesztből kivágott részből, a „Bounding Box”-ot a találatból, és a „status” false értékét a hibavezetékből. Eredménytelen keresés vagy hiba bekövetkezte esetén a kép egy üres konstans lesz, a „Bounding Boxot” a referenciadarab „Global Rectangle” tagjából hozzuk létre, és a „status” true értéket vesz fel.



32. ábra: TesztdarabKeres VI

4.6.4.3. Elemzés

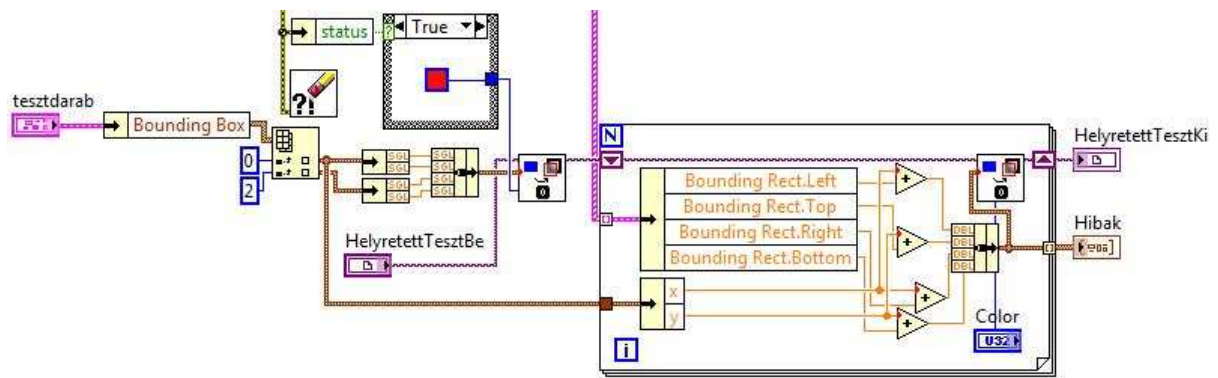
Az „Elemesz” VI megkapja a referencia- és tesztdarabok előkészített, bináris változatát. Első lépésben a „BinárisKivonas” VI meghatározza a kettő közti eltéréseket. A különbségeket tartalmazó kép még tele lehet hibának nem minősülő, 1-2 pixel széles részecskékkel, amik a két kép még mindig nem tökéletes fedéséből, valamint a fényviszonyokból adódnak. Az „IMAQ Skelton” segítségével meghatározzuk a részecskéknek a „váz-szerkezetét”. Az elcsúszásból adódó eltérések esetén a részecske és a „skelton” egy és ugyanaz. Ezek eltűnnek, ha a különbséget tartalmazó képből kivonjuk a „skelton-képet”, viszont ezáltal a nagyobb objektumok „szétesnek”. Ekkor van szükség a morfológiai eljárások közül a bezárásra, amely összekapcsolja az egymáshoz közel elhelyezkedő részecskéket. Végezetül az „IMAQ Particle Filter 3” használatával eltüntetjük a még mindig jelenlevő apró zajokat. Beállításait az alkalmazástól függően a felhasználó adhatja meg. Azok az objektumok, amik mindezek után a képen maradnak, hibának tekintendők, és az „IMAQ Particle Analysis Repor” gyűjti össze őket.



33. ábra: Elemesz VI kiértékelés része

A tesztdarab „Bounding Box” eleméből „Rectangle”-t készítünk, és ezt „Overlay”-ként megjelenítjük a tesztképen. Az „Overlay” színe attól függ, hogy történt-e hiba a feldolgozás során.

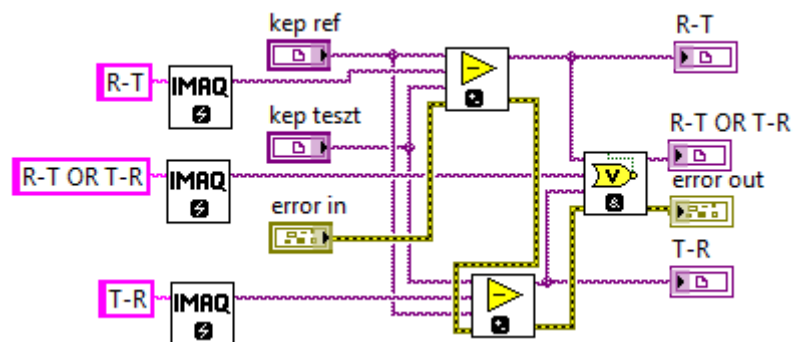
A „Particle Reports” tömbön For-ciklus segítségével végigmegyünk. Azt szeretnénk tudni, hogy a nagy képen hol találhatóak a hibák. Viszont a „Particle Report” a hibának minősülő részecske elhelyezkedését a teszt darabon érti. Az ezt megadó „Bounding Rect” tagjaihoz hozzáadva a tesztdarabhoz tartozó „Bounding Box” kezdő x és y koordinátáit, megkapjuk a hibák tényleges helyét. Ezt a képen megjelenítjük, és „Hibák” néven a kimentre küldjük.



34. ábra: Elemesz VI eredmény összegző része

4.6.4.3.1. Bináris kivonás

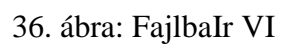
A VI két bináris képet kap, amiknek az összevetését végzi el. Első lépésbe az „IMAQ Subtract” alkalmazásával a referencia képből kivonjuk a tesztet. Eredményként azok az elemek maradnak, amik az első képen szerepeltek, de a másodikon nem. Ezután a tesztképből vonjuk ki a referenciát. Végül a két kapott képet logikai „VAGY” operátorral („IMAQ OR”) egyesítjük. Ez a két bemeneti kép összes eltérését mutatja. A három létrehozott képet külön kimenetekre küldjük.



35. ábra: BinárisKivonas VI

4.6.4.4. Fájlba írás

Az eredmények fájlba írása „Format Into File” segítségével történik. Az „Elemesz” VI-tól kapott, hibákat tartalmazó tömbön egy For-ciklussal végigmegyünk. A csoportot az Unbundle bontja szét, a kapott elemeket egymás mellé írjuk, a sor végére End of Line Constant kerül.



Összefoglalás

Az elektronikai eszközök alkatrészeinek összeillesztése forrasztás útján történik. Azért ez a legalkalmasabb módszer, mert vele egy erős, tartós kötés hozható létre, úgy, hogy közben a felület sértetlen marad. A gyártósoron felületszereléshez újraömllesztéses módszert, furatszereléshez pedig hullámforrasztást alkalmaznak. A folyamat összetettségéből kifolyólag bárhol történhet hiba. A forrasztások jelentős hibáit két nagy csoportra oszthatjuk, hiányra és többletre. A gyártók a megtermelt áru mennyisége mellett nagy hangsúlyt fektetnek a minőségre is. A technika fejlődése során az alkatrészek mérete egyre jobban lecsökken, integráltságuk foka pedig folyamatosan növekszik, így az ember által végzett vizsgálat nem alkalmazható kellő hatékonysággal. Az automatizált tesztelés megvalósítására többféle berendezés áll rendelkezésre. Az optikai vizsgálat előnye, hogy nem igényel érintkezést a vizsgált elemmel, így megkíméli azt a fizikai behatásoktól. A rendszert számítógép vezérli. Ezen kapnak helyet a vizsgálatához szükséges fájlok (pl. CAD-állományok), valamint ide lesz lementve az ellenőrzés eredménye is. A rendszer másik fontos eleme az optikai fej, melyen a kamera és a világítás található. Forrasztások vizsgálatához a sötétlátóteres, egyenletes fény a legalkalmasabb. Fényforrásként általában LED-et alkalmaznak, annak hosszú élettartama és egyéb kiváló tulajdonságainak köszönhetően. A fényerő és a megvilágított terület növelése érdekében több LED-et építenek egy csoportba. Az ellenőrzőprogram képfeldolgozási eljárások segítségével végzi a vizsgálatot. Az időben észrevett hibás alkatrész még viszonylag könnyen javítható.

A forrasztások vizsgálatának kapcsán próbáltam megérteni a hibák kialakulásának okait, valamint körülményeit a különböző forrasztási technikák esetén. Igyekeztem a tesztelő berendezéseket minél szélesebb körben tanulmányozni, és összehasonlítani alkalmazhatóságuk, illetve elhelyezésük szerint. Külön figyelmet fordítottam az optikai vizsgálaton alapuló rendszerekre. Működésük mélyebb megismerése céljából készítettem egy programot, mely ellenőrzi a forrasztások kötéseit. A program elkészítésére LabVIEW8.6-ot használtam, mivel az kifejezetten mérési és vizsgálati rendszerek kialakítására készült. Ezen túlmenően pedig támogatja a képfeldolgozást. A többféle vizsgálati módszer közül a képösszehasonlításra alapuló eljárást kiviteleztem. A program fejlesztése közben személyesen is tapasztaltam, hogy egy ilyen rendszerhez a legkiválóbb eszközökre van szükség. A felvételek készítéséhez használt kamerához tartozó objektív elég erősen torzít.

Ilyen körülmények között pedig még azokat a képeket sem lehet összevetni, melyek ugyanarról a panelről készültek. A felvétel rossz minőségéből adódó problémákat szoftveresen kell kiküszöbölni. A program készítése során elsajátítottam a képfeldolgozási eljárások gyakorlati használatát. Igyekeztem a mögöttük rejlő matematikai összefüggések megismerése révén teljesebb képet kapni működésükről.

A program által végzett vizsgálat képösszehasonlításon alapul. A referencia és a tesztelni kívánt panelekről felvételt készítünk. Megnyitjuk a referencia és a teszt képet. A kamera torzításából adódó hibákat kalibrációval korrigálhatjuk. Ehhez szükség van egy rácsot (grid-et) tartalmazó képre. Kalibráció során, a rács alapján számolja a program a torzulás mértékét, majd az eredménytől függően javítja a képet. Ha referencia és a tesztkép a felvételen más-más pozícióban helyezkedik el, az összehasonlítást nem lehet elvégezni. Ebben az esetben a tesztképet a referenciához igazítjuk. A képek éleihez koordináta-rendszert illesztünk. A koordináta-rendszerek kiindulási pontjai megadják az eltolást x és y irányban, a szögük pedig a két kép elfordulását egymáshoz képest. Ezen adatok alapján a tesztképet már pozícionálni tudjuk. Következő lépésként a képeket részekre osztjuk. Erre azért van szükség, mert kisebb területen gyorsabban elvégezhetőek a képfeldolgozási műveletek, másrészt ez is segíti a pontosabb illeszkedést. A tesztdarabokat a referenciadarabok alapján keressük mintaillesztés segítségével. Tulajdonképpen a darabokat fogjuk egymáshoz viszonyítani, és nem az egész képet. A forrasztások kiemelése érdekében elvégzünk egy Diszkrét Koszinusz Transzformáción alapuló szűrést, majd egy alacsony küszöbérték mellett binarizáljuk a képet. A bináris képen már elvégezhető az összehasonlítás. A referenciából kivonva a tesztet egy pozitív képet kapunk, a tesztből kivonva a referenciát pedig egy negatívot. A keletkezett képek még tartalmazhatnak zajokat, nem kívánt információkat, melynek oka lehet pl. a még mindig nem tökéletes fedés, vagy egyéb fényvisszaverődésből adódó „zajok”. Elkészítjük a pozitív és negatív kép részecskék váz-szerkezetét (skelton), melyet kivonunk a megfelelő pozitív és negatív képből. Ezáltal eltűnnek a vékony objektumok, a szélesebbek pedig szétesnek. Hogy ezt megszüntessük, egy morfológiai bezárást kell végrehajtani. Az esetlegesen még mindig jelenlévő 1-2 pixelnyi zajokat részecskeszűréssel eltávolítjuk. Ami a képen marad, azt kell hibának tekinteni. A program ezeket összegyűjti, a kijelzőn megjeleníti és egy fájlba menti.

A szakdolgozatom írásánál elsősorban angol szakkönyvekre és, magyar nyelvű egyetemi segédanyagokra támaszkodtam. A világítástechnika témakörében a különböző gyártók termékleírásaiban ismertették az egyes típusok alkalmazási területeit. Az optikai forrasztávizsgálatok háttéréről csak két angol nyelvű tanulmányban volt utalás.

Köszönetnyilvánítás

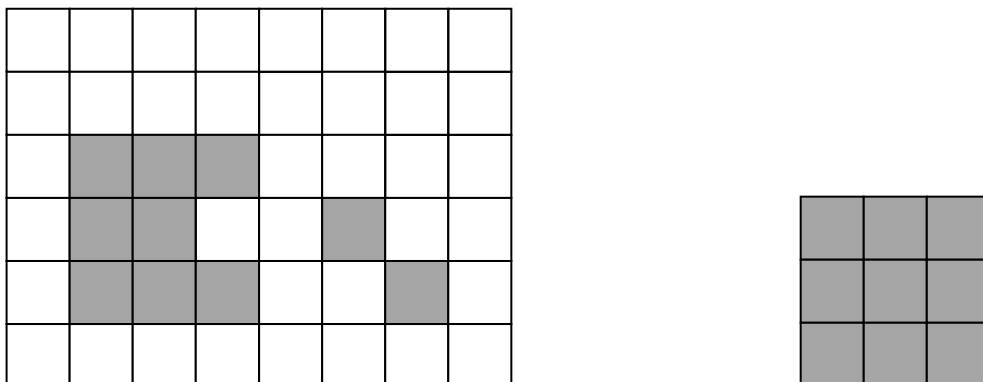
Köszönettel tartozom Dr. Cserhádi Csabának, amiért elvállalta a témavezetést, és ezen túlmenően a sok segítségéért.

Köszönöm Vámos Dánielnek a program készítése közben nyújtott tanácsait.

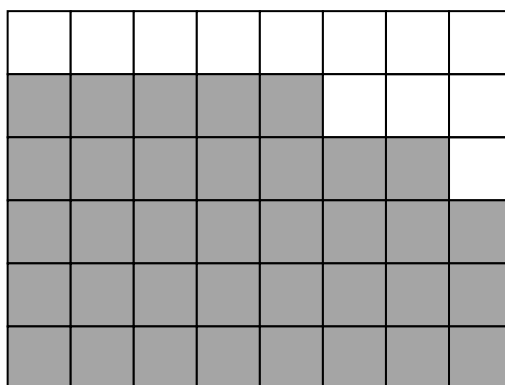
Köszönettel tartozom a DE TTK Szilárdtestfizika Tanszék munkatársainak és tanszékvezetőjének Dr. Szabó Istvánnak.

Függelék

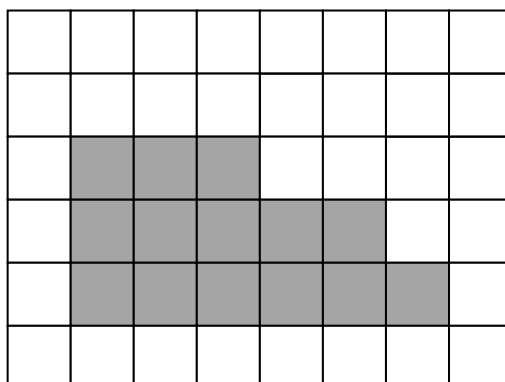
1.) Morfológiai bezárás megvalósítása



37. ábra: kép és a szerkezeti elem

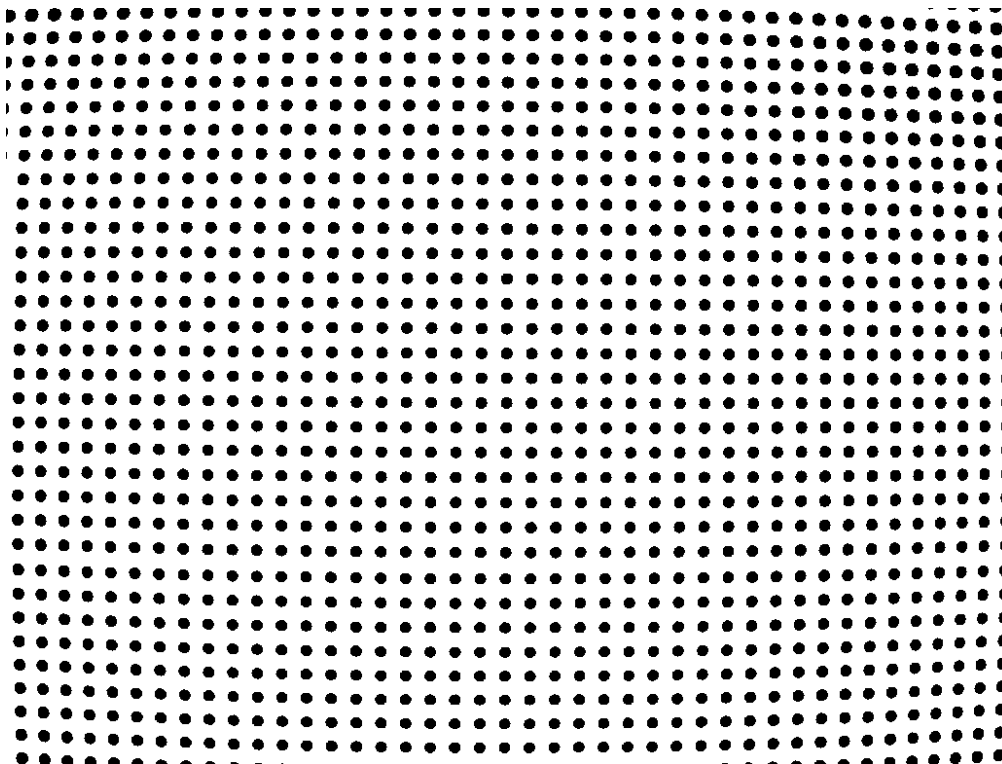


38. ábra: a kép nyújtása a szerkezeti elemmel

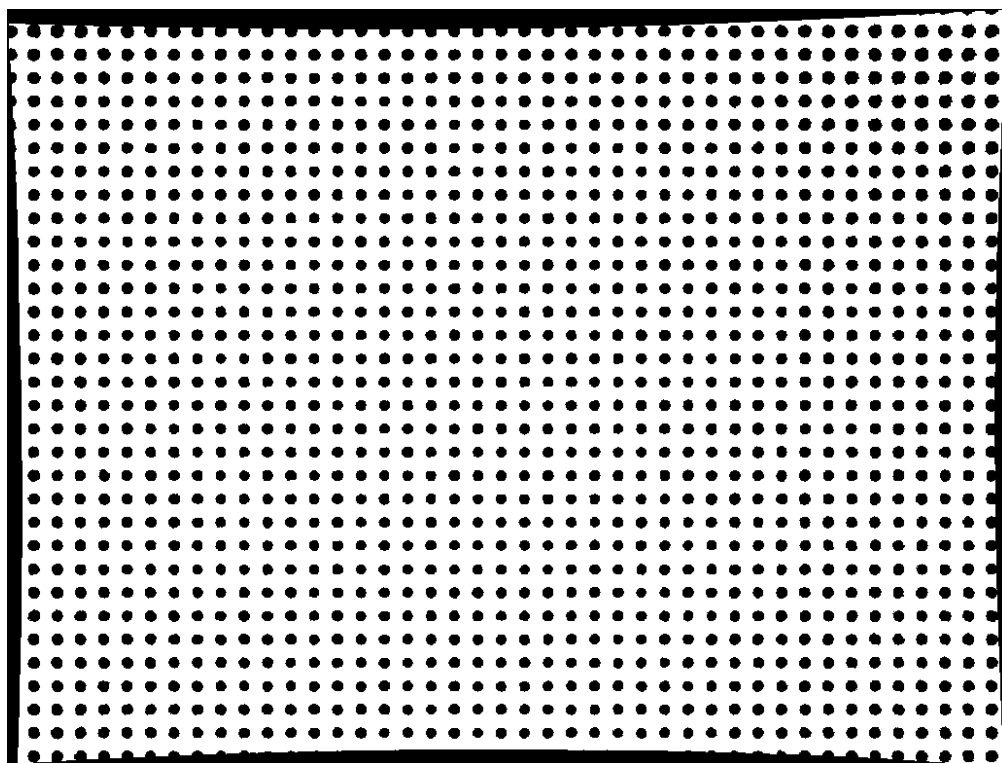


39. ábra: a nyújtott kép erodálása

2.) Kalibrálás eredménye

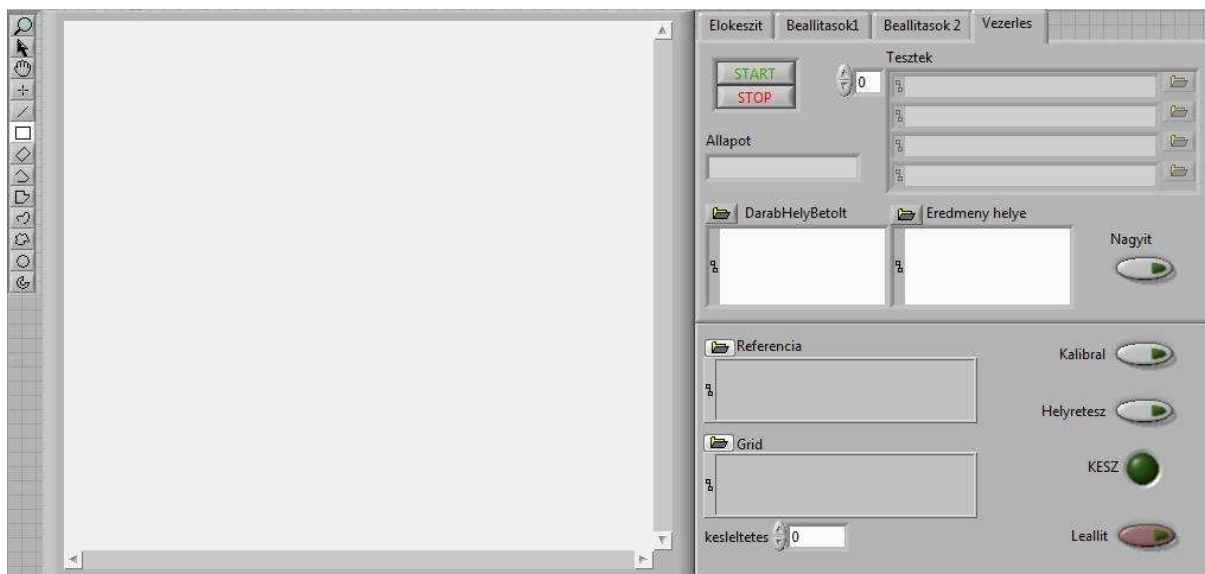


40. ábra: a kamera által készített rács torzulása

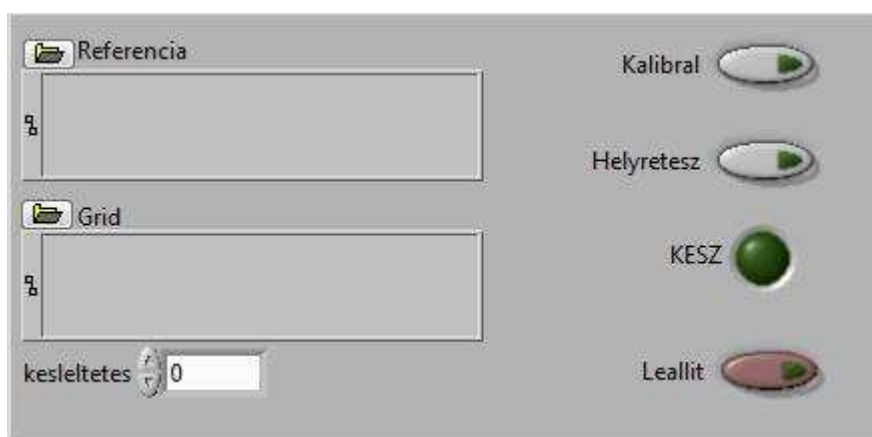


41 ábra: rács a kalibráció elvégzése után

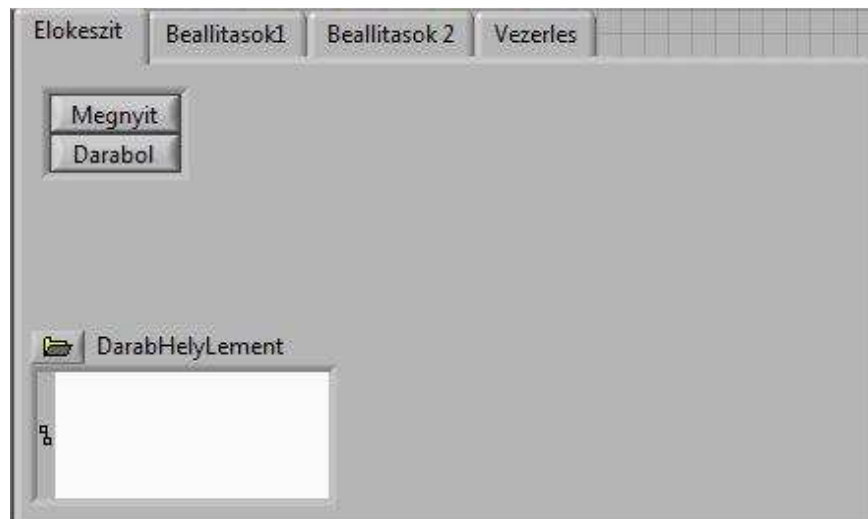
3.) A program kezelőfelülete



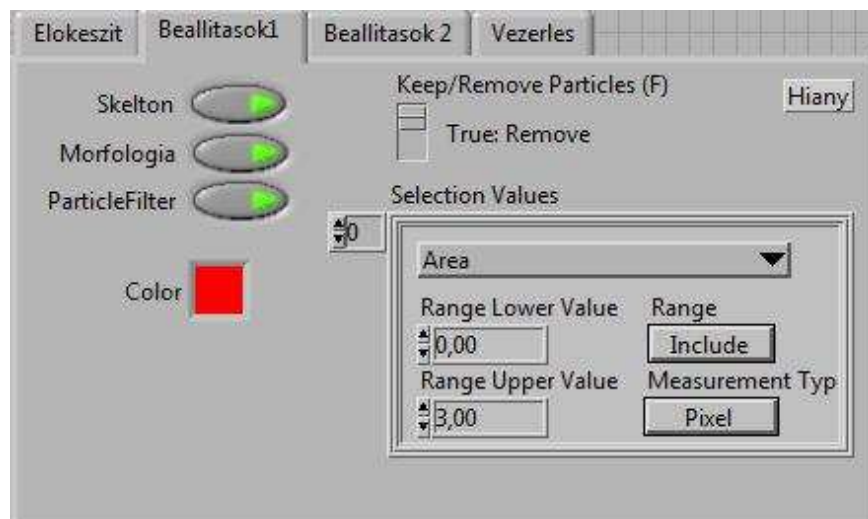
42.ábra: a program kijelzője és kezelőfelülete



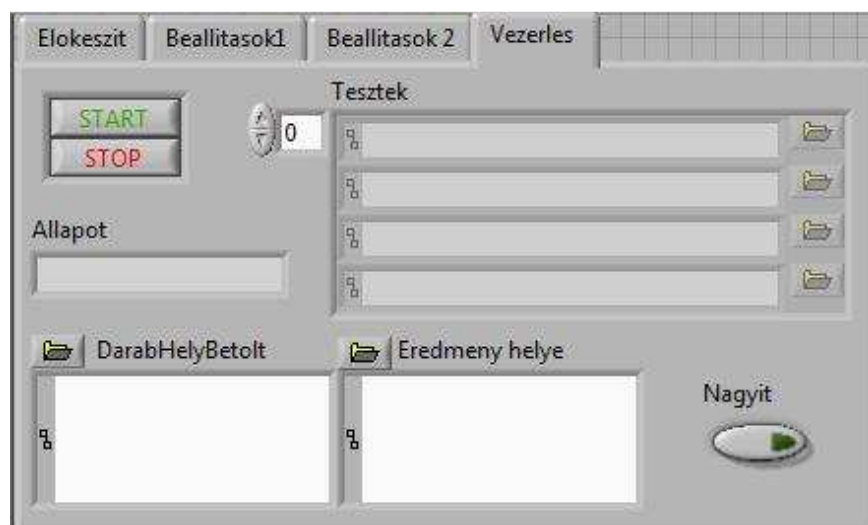
43. ábra: a kezelőfelület állandó része



44. ábra: az „Elokeszit” lap



45. ábra: a „Beallitasok” lap



46. ábra: a „Vezertes” lap

Irodalomjegyzék

- [1] <http://www.uni-miskolc.hu/~wwwfemsz/forgacs.htm>
- [2] Mike Judd, Keith Brindley, Soldering in Electronics Assembly, 1999.
- [3] <http://www.cnctar.hunbay.com/G-vox/forraszt%E1stechnika/forraszt%E1stechnika%20+.ppt>
- [4] Regős Péter, Az ólommentes forrasztás, Budapesti Műszaki Egyetem, 2004.
- [5] Czap László, Képfeldolgozás, Miskolci Egyetem
- [6] http://www.radio-electronics.com/info/t_and_m/ate/automatic-test-equipment-basics.php
- [7] http://www.radio-electronics.com/info/t_and_m/ate/aoi-automatic-automated-optical-inspection.php
- [8] <http://www.smt.hupont.hu/6/aoi>
- [9] Dr. Pinkola János, Gordon Péter, Dr. Gál László, Illés Balázs, Krammer Olivér, Kovács Róbert, Dr. Ruzinkó Miklós, Janóczki Mihály, Berényi Richárd, Balogh Bálint, Mérési útmutatók, Technológiai Folyamatok és Minőségellenőrzésük Laboratórium, Budapesti Műszaki és Gazdaságtudományi Egyetem, 2008.
- [10] Janóczki M., Automatikus optikai vizsgálat (1. rész). ELEKTRONET 18:(6) pp. 18-19. (2009)
- [11] <http://www.mair-elektronik.de/pdf/AOI-Automatische%20optische%20Inspektion.pdf>
- [12] Daryl Martin, Vision Lighting Seminar, 2005.
- [13] RVSI NER, Illumination for Machine Vision Guide
- [14] NERLITE, Precise Lighting for Machine Vision, Brochure September 2007.
- [15] Sarp Ertürk, Digital Image Processing, 2003.
- [16] National Instruments Corporation, IMAQ Vision Concepts Manual, 2005.
- [17] http://e-oktat.pmmf.hu/kepeshang_4_fejezet
- [18] <http://web.fmt.bme.hu/subjects/dip/eszkoz2.pdf>
- [19] Batta Imre, Mintavétel elmélete, BME Építészmérnöki Kar, 2008.
- [20] Csiszár János, Az adatredukció alapjai digitális hang és képanyagok esetében, 2009.
- [21] <http://zone.ni.com/reference/en-XX/help/371361E-01/lvanls/dct/>
- [22] http://e-oktat.pmmf.hu/bevezetes_grafikus_programozas
http://e-oktat.pmmf.hu/2_fejezet_grafikus_programozas
http://e-oktat.pmmf.hu/3_fejezet_grafikus_programozas
http://e-oktat.pmmf.hu/4_fejezet_grafikus_programozas
http://e-oktat.pmmf.hu/6_fejezet_grafikus_programozas

^[23] Zuwairie Ibrahim, Syed Abdul Rahman Al-Attas, Wavelet-Based Printed Circuit Board Inspection System, International Journal of Signal Processing 1;2; 2005.

^[24] N.S.S. Mar*, C. Fookes, P.K.D.V. Yarlagadda, Design of automatic vision-based inspection system for solder joint segmentation, Journal of Achievements in Materials and Manufacturing Engineering ,Volume 34, Issue 2, June 2009